ISSN 2788-5518

# ІНФОКОМУНІКАЦІЙНІ ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

**№ 2 (06), 2023**

**НАУКОВИЙ ЖУРНАЛ**

Видавничий дім
«Гельветика»
2023

ISSN 2788-5518

# INFOCOMMUNICATION AND COMPUTER TECHNOLOGIES

№ 2 (06), 2023

SCIENTIFIC JOURNAL

Publishing House
“Helvetica”
2023

**Засновник, видавець та виробник журналу:** «Відкритий міжнародний університет розвитку людини «Україна».

**Свідоцтво про державну реєстрацію:** Серія КВ № 24782-14722 Р від 06.04.2021 р.

**Журнал включений до переліку наукових фахових видань ВАК України** категорія «Б», додаток 4 до наказу МОН України № 735 від 29.06.2021 р. та додаток 2 до наказу МОН України № 320 від 07.04.2022 р.

**Журнал відповідає вимогам наказу МОН України № 32 від 15.01.2018 р.**

**Має міжнародні коди: ISSN 2788-5518 та DOI 10.36994/2788-5518.**

**Журнал індексується у міжнародних базах та каталогах Google Scholar, Index Copernscus, Vernadsky National Library of Ukraine.**

**Мова видання:** українська, англійська

Рекомендовано до друку вченою радою відкритого міжнародного університету розвитку людини «Україна» (протокол № 10 від 28.12.2023 р.)

**URL-адреса видання: visn-icct.uu.edu.ua/**

**Тематика журналу:**

- Теоретичні основи інфокомунікацій та комп’ютерних наук
- Інфокомунікаційні системи
- Безпроводові технології
- Технічна електродинаміка та антени
- Волоконно-оптичні системи
- Радіорелейні та супутникові системи
- Телевізійні системи
- Системи відеоконференцзв’яку та відеоспостереження
- Комп'ютерні системи та мережі
- Комп'ютерна і програмна інженерія
- Системи SCADA
- Захист інформації та кібербезпека
- Прикладна математика та моделювання
- Метрологія та інформаційно-вимірювальні системи
- Медична та екологічна електроніка
- Автоматизаія управління технологічними процесам

В журналі публікуються статті для захисту дисертацій зі спеціальностей: 121 Інженерія програмного забезпечення. 122 Комп’ютерні науки. 123 Комп’ютерна інженерія. 125 Кібербезпека та захист інформації. 151 Автоматизація та комп’ютерні інтегровані технології. 152 Метрологія та інформаційно-вимірювальна техніка. 172 Телекомунікації та радіотехніка

**Founder, publisher and producer of the journal: "Open International University of Human Development" Ukraine".**

**Certificate of state registration:** Series KV № 24782-14722 R from 06.04.2021.

**The journal is included in the list** of scientific professional publications of the Higher Attestation Commission of Ukraine – category "B", Annex 4 to the order of the Ministry of Education and Science of Ukraine № 735 from 29.06.2021 and Annex 2 to the order of the Ministry of Education and Science of Ukraine № 320 from 07.04.2022

**The journal meets the requirements of the order of the Ministry of Education and Science of Ukraine № 32 from 15.01.2018.**

**It has international codes: ISSN 2788-5518 and DOI 10.36994 / 2788-5518.**

**The journal is indexed in international databases and directories of Google Scholar, Index Copernicus, Vernadsky National Library of Ukraine**

**Language of publication:** Ukrainian, English

Recommended for publication by the Academic Council of the Open International University of Human Development "Ukraine" (Protocol № 10 dated 28.12.2023.)

**Publication URL: visn-icct.uu.edu.ua/**

**Subject journal:**

- Theoretical foundations of infocommunications and computer science
- Infocommunication systems
- Wireless technology
- Technical electrodynamics and antennas
- Fiber optic systems
- Radio relay and satellite systems
- Television systems
- Video conferencing and video surveillance systems
- Computer systems and networks
- Computer and software engineering
- SCADA systems
- Information security and cybersecurity
- Applied mathematics and modeling
- Metrology and information-measuring systems
- Medical and environmental electronics
- Automation of process control

The journal publishes articles for the defense of dissertations in the following specialties: 121 Software engineering. 122 Computer science. 123 Computer engineering. 125 Cybersecurity. 151 Automation and computer integrated technologies. 152 Metrology and information and measurement technology. 172 Telecommunications and radio engineering

# ЗМІСТ

## ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ



## КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

# CONTENTS

## INFOCOMMUNICATION TECHNOLOGIES



## COMPUTER TECHNOLOGIES

# ІНФОКОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ

**УДК 004.056 (083.94)**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-01**

## ОЦІНКА ЗАХИЩЕНОСТІ БЕЗДРОТОВИХ МЕРЕЖ У МІСЬКОМУ СЕРЕДОВИЩІ З УРАХУВАННЯМ ВИКОРИСТАННЯ ПРОТОКОЛІВ БЕЗПЕКИ

***Гальчинський Л. Ю.,*** *канд. техн. наук, доцент, Національний технічний університет України «Київський політехнічний інститут імені Ігора Сікорського», Київ, Україна*
*https://orcid.org/0000-0002-3805-1474, hleonid@gmail.com*

***Корольова В. Р.,*** *Національний технічний університет України «Київський політехнічний інститут імені Ігора Сікорського», Київ, Україна*
*korolova.workspace@gmail.com*

***Анотація.*** *У цій роботі представлені результати дослідження оцінювання рівня захищеності бездротових мереж у міському середовищі на прикладі міста Києва. Здійснено аналіз чинників, що впливають на рівень безпеки, зокрема шифрування, автентифікації та цілісності, які впливають на рівень механізму захисту трафіку бездротових мереж, зокрема типів протоколів WEP, WPA, WPA2 та WPA3, залежно від моделей атак. Показані вразливості протоколів у контексті різновидів конфігурацій протоколів. Визначена перевага з точки зору безпеки одного типу протоколу стосовно іншого. На основі даних публічних БД визначено актуальну кількість бездротових мереж і відповідні частки протоколів безпеки у світовому масштабі. Було проведено автоматизоване дослідження безпеки бездротової мережі на основі методології Wardriving у міському середовищі Києва, за допомогою якої в пасивному режимі були отримані дані типів протоколів безпеки. Результати сканування показали відсутність використання як застарілого протоколу WEP, так і новітнього протоколу WPA3. За результатами обробки даних сканування були визначені частки різних типів протоколів безпеки. Причому методика сканування дала змогу отримати не тільки ширший спектр типів протоколів безпеки, ніж класичні значення WEP, WPA, WPA2 чи WPA3, а й варіанти використання стандартів, які забезпечують шифрування, автентифікацію та цілісність. На основі двоетапної процедури було побудовано інтервальну шкалу оцінювання рівня безпеки кожного з типів протоколів. На першому етапі протоколи були згруповані за певними рівнями захисту, а потім проранжовані в межах відведеного інтервалу за перевагами використаних стандартів, завдяки чому було отримано порядкову шкалу оцінки рівня безпеки типу протоколу. Далі експертно були визначені оцінки кожного типу протоколу, що дає змогу здійснювати перехід до інтервальної шкали оцінювання. На основі цієї шкали та визначених часток типів протоколу за цією вибіркою сканувань отримано оцінку рівня захищеності мереж у міському середовищі Києва.*

***Ключові слова:*** *бездротові мережі; рівень безпеки; протоколи безпеки; типи протоколів безпеки; сканування мережі Wardriving; методика оцінювання.*

## ASSESSMENT OF THE SECURITY OF WIRELESS NETWORKS IN AN URBAN ENVIRONMENT, TAKING INTO ACCOUNT THE USE OF SECURITY PROTOCOLS

***Leonid Galchynsky,*** *Ph. D Assoc. National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*
*https://orcid.org/0000-0002-3805-1474, hleonid@gmail.com*

***Valeriia Korolova,*** *National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*
*korolova.workspace@gmail.com*

***Abstract.*** *This paper presents the results of a study of assessing the level of security of wireless networks in an urban environment on the example of Kyiv. An analysis of factors affecting the level of security, in particular encryption, authentication and integrity, which affect the level of the mechanism for protecting wireless network traffic, in particular the types of protocols WER, WPA, WPA2 and WPA3, depending on the attack models, is carried out. Protocol vulnerabilities are shown in the context of different protocol configurations. The security advantage of one type of protocol over another is determined. Based*

*on the data from public databases, the current number of wireless networks and the corresponding shares of security protocols on a global scale are determined. An automated study of wireless network security based on the Wardriving methodology in the urban environment of Kyiv was conducted, which was used to obtain data on the types of security protocols in a passive mode. The scan results showed the absence of the use of both the outdated WEP protocol and the newest WPA3 protocol. As a result of scan data processing, the shares of different types of security protocols were determined. Moreover, the scanning methodology allowed us to obtain a wider range of security protocol types than the classic values of WEP, WPA, WPA2 or WPA3, and options for using standards that provide encryption, authentication and integrity. Based on a two-stage procedure, an interval scale for assessing the security level of each type of protocol was built. At the first stage, the protocols were grouped by certain levels of protection, and then ranked within the allocated interval according to the advantages of the standards used, which allowed us to obtain an ordinal scale for assessing the security level of a protocol type. Next, the expert assessment of each type of protocol was determined, which allows for the transition to an interval assessment scale. Based on this scale and the determined shares of protocol types in this sample of scans, an assessment of the level of security of networks in the urban environment of Kyiv was obtained.*

***Key words:*** *wireless networks; security level; security protocols; types of security protocols; Wardriving network scan; assessment methodology.*

**Вступ**

Бездротова мережа, яка зараз відома широкому загалу як Wi-Fi, – це скорочення від Wireless Fidelity. Стандарт, що дає змогу отримати доступ до мережі без використання будь-яких кабелів, тобто можливість передавати дані за допомогою невидимих радіохвиль. Він ґрунтується на стандарті IEEE 802.11, впровадженому в 1991 році, хоча практичну реалізацію цієї технології було здійснено лише в 1997 році. Впровадження стандартів IEEE 802.11 WLAN у 1997 році сприяло зростанню популярності WLAN, який став повсюдним рішенням щодо підключення для багатьох користувачів по всьому світу [1]. На рис. 1а та рис. 1б показані ключові елементи WLAN:

Рис. 1а. Інтерфейс бездротових мереж

Рис. 1б. Бездротова точка доступу

Започаткування ери бездротових мереж ініціювало перехід до нового формату використання мережі, що полягало в значних перевагах завдяки низькій затримці та високій швидкості під час передачі даних, зручному налаштуванню, мобільності, масштабованості, відсутності кабелів і можливому підключення з різних регіонів світу, що призвело до фантастичного зростання мереж Wi-Fi.

Розвиток мереж Wi-Fi забезпечують дві організації: IEEE та Wi-Fi Alliance®. IEEE розробляє стандарти, які описують процедури, обмеження, значення й алгоритми для встановлення з'єднання WLAN. Wi-Fi – це торговельна марка, що належить Wi-Fi Alliance, яка за допомогою ретельних тестів підтверджує взаємодію між усіма пристроями із цим знаком і надає сертифікати розробникам і виробникам обладнання.

Відповідно до дослідження, проведеного Wi-Fi Alliance®, глобальна економічна вартість Wi-Fi у 2023 році оцінюється в понад 3,5 трильйона доларів США. Очікується, що до 2025 року ця вартість зросте майже до 5 трильйонів доларів [2]. Дані в мережах Wi-Fi передаються за аналогічним протоколом TCP/IP, як і у разі передачі по кабелю. Проте середовище, яким передаються пакети в мережевій комунікації, суттєво відрізняється. Бездротові мережі для зв'язку між пристроями в домашніх і корпоративних мережах використовують радіодіапазони 2,4 та 5 ГГц. Рішення цього типу вважаються набагато менш безпечними, ніж стандартні кабельні або оптичні мережі. Неавторизовані користувачі можуть розмістити станції для несанкціонованого прихованого прослуховування. Таке середовище породжує ряд критичних вразливостей, які призводять до суттєво негативних наслідків. На перших етапах розвитку бездротових мереж питання про безпеку даних не було актуальним, бо цільова аудиторія була занадто вузькою, а вартість послуг – занадто високою, тому шифрування даних та автентифікація сутностей, які здійснювали підключення до мереж, практично не передбачались. Хоча деякі механізми безпеки Wi-Fi, створені для захисту бездротової локальної мережі, передбачалися стандартом IEEE 802.11. Зокрема, це ідентифікатор набору послуг (SSID), який використовується для контролю доступу до точки доступу (AP), список контролю доступу (ACL) для запобігання неавторизованому доступу, а також алгоритм протоколу Wired Equivalent Privacy (WEP). Проте WEP був визначений стандартом як необов'язковий, тому ні точки доступу, ні бездротові пристрої не зобов'язані його використовувати [3].

У такому стані система Wi-Fi не забезпечувала захист конфіденційності, цілісності та надійності даних, що передавались мережею. Зловмисники могли збирати, аналізувати, модифіковувати вміст перехоплених пакетів, у яких могли міститися критичні дані користувачів. З поширенням попиту й удосконаленням інфраструктури бездротових мереж постали питання про захист даних.

Реальне використання WEP почалося тільки через два роки, після впровадження стандарту IEEE 802.11 [4]. Як наслідок цих викликів, організацією Wi-Fi Alliance у 1999 році був доопрацьований до практичного використання перший протокол безпеки WEP (Wired Equivalent Privacy). Цей протокол мав 64/128-бітне шифрування з використанням алгоритму RC4. Проте із часом виявилися слабкі сторони WEP, а саме вразливість у побудові протоколу безпеки стосовно статичних ключів шифрування, вектора ініціалізації, що призвело до реалізації численних кібератак. Для вирішення проблеми безпечного використання мереж 2003 року було представлено нове покоління протоколів – WPA, яке ґрунтувалось на подальшій підтримці у використанні алгоритму RC4, зі 128-бітним шифруванням і новим протоколом цілісності тимчасового ключа в протоколі захищеного доступу TKIP. WPA вважався програмною обгорткою WEP, яка вирішувала проблеми та вразливості, виявлені в попередньому протоколі безпеки. Згодом з'ясувалося, що протокол WPA теж ненадійний через існування автентифікації PSK, технології WPS, тому 2004 року з'явилась оновлена версія протоколів безпеки – WPA2, з наявним алгоритмом AES і режимом лічильника CCMP (Cipher Block Chaining Message Authentication Protocol). WPA2 став гарантом безпеки бездротових мереж, пропагував надійність і постійну підтримку через оновлення, які регулярно впроваджувалися для покращення захисту. Протокол WPA2 вирішував недоліки попередніх протоколів, інтегрував функції захисту та досяг покращення режимів безпеки Personal та Enterprise. У 2006 році організація Wi-Fi Alliance оголосила, що технологія безпеки шифрування WPA2 має бути обов'язковою функцією для всіх нових Wi-Fi CERTIFIED-продуктів, оскільки відсутність цієї сертифікації сповільнюватиме пристрої, мінімізуватиме широкий спектр додаткового функціонала та технічної підтримки.

На поточний момент протокол WPA2 продовжує залишатися популярним у середовищі користувачів, незважаючи на вразливості автентифікації 4-way handshaking, що призвели до імплементації однієї з основних атак – KRACK (2017 рік), за допомогою якої зловмисники можуть перехоплювати мережеві пакети, зчитувати й модифіковувати дані. Розробники випускали патчі, щоб боротись з такими атаками, проте остаточно цю проблему не вирішили.

Із метою подолання вразливості KRACK та ряду інших вразливостей у 2018 році Wi-Fi Alliance було прийнято рішення про впровадження наступного покоління протоколів безпеки – WPA3. У цьому протоколі використовується 128/192-бітне шифрування, протокол автентифікації SAE, вдосконалені режими WPA3-Personal, WPA3-Enterprise. Створено нові технології підключення до бездротових мереж, як-от Enhanced Open та Easy Connect.

Незважаючи на очевидні переваги, з 2018 року і дотепер реалізації WPA3 не довелося стати повноцінним стандартом сертифікованого апаратного забезпечення. Причиною цього стали як приховані нові вразливості цього протоколу, так і складнощі його впровадження в реальних мережах Wi-Fi, у яких використовуються різнотипні протоколи захисту даних. Отже, сучасні мережі Wi-Fi функціонують у стані використання різнотипних протоколів безпеки. Найбільш масштабні Wi-Fi мережі сконцентровані в міських агломераціях. Тож виникають питання про рівень захищеності таких мереж як загалом, так і для окремо взятого користувача.

**Виконання досліджень**

Спочатку проаналізуємо характеристики протоколів безпеки, які використовують користувачі бездротових мереж.

**Механізм Wired Equivalent Privacy (WEP).** Назва протоколу Wireless Equivalent Privacy – WEP вказує на наміри розробників [5] зробити безпеку бездротових мереж, еквівалентну рівню безпеки, яку забезпечують дротові локальні мережі. Незабаром практика показала, що рівень безпеки, який пропонує WEP, не став цільовим, хоча це також був значний еволюційний крок порівняно з відкритим доступом. Проте, незважаючи на виявлені пізніше недоліки протоколу, WEP забезпечує певний рівень безпеки, наприклад може стримувати випадкове стеження. Деякі користувачі вважають цей рівень безпеки достатнім, навіть протягом чверть століття після появи цього протоколу.

Коли WEP активний, кожен пакет 802.11 шифрується окремо, як потік шифру RC4, згенерований 64-бітним ключем RC4. RC4 – це симетричне кодування, яке складається з двох частин: алгоритму планування ключів (KSA), який отримує початкову перестановку з ключа випадкового розміру (типовий розмір становить 40–256 біт) і алгоритму псевдовипадкової генерації (PRGA), який використовує початкову перестановку для генерації псевдовипадкової вихідної послідовності (ключ RC4). Цей ключ складається з 24-бітного вектора ініціалізації (IV) і 40-бітного ключа WEP. Зашифрований пакет генерується за допомогою побітового виключного АБО (XOR) вихідного пакета та потоку RC4. IV вибирається відправником і може періодично змінюватися, щоб кожен пакет не шифрувався повторним потоком шифру. Вектор ініціалізації надсилається у відкритому вигляді з кожним пакетом. Додаткове 4-байтове значення перевірки цілісності (ICV) обчислюється на вихідному пакеті та додається. ICV також шифрується за допомогою потоку шифрів RC4.

Проте цього виявилося недостатньо. Однією з найслабших частин WEP є 24-бітний вектор ініціалізації, який може призвести до повторного використання потоку ключів. Повторне використання потоку ключів, зі свого боку, забезпечує успішні атаки криптоаналізу на зашифрований текст. Отже, WEP було визнано нездатним у досягненні цілей безпеки та таким, який містить серйозні недоліки безпеки [6]. Іншою вразливістю було використання алгоритму CRC-32 для перевірки цілісності. CRC не є криптографічно надійним і поступається для використання більш надійним засобам, таким як хеш-функції. Слабкість також була виявлена в механізмі планування ключів RC4. Шляхом ідентифікації великої кількості слабких ключів, а також через те, що перший байт, згенерований RC4, містить інформацію про окремі ключові байти, можна проаналізувати достатню кількість пакетів, зашифрованих WEP, тобто відновити секретний ключ у WEP [7]. Крім того, були виявлені ще деякі слабкості, зокрема використання одного ключа WEP, який спільно використовується для кожного вузла в мережі. Оскільки зміна ключів є виснажливим процесом, системні адміністратори рідко змінюють їх.

За декілька років як користувачам, так і експертам стало зрозуміло, що WEP – далеко не ідеальне рішення безпеки. Проте його використовували і продовжують надалі використовувати. Частина користувачів Wi-Fi не надто переймається проблемами захисту, хоч і вважає, що краще мати якийсь механізм безпеки стримування, ніж не мати жодного. Так, зловмисник може зламати ключі WEP за наявності часу та достатньо слабких ключів, але це не є приводом уникати можливого захисту.

Однак значна частина користувачів була надто стурбована, і це спонукало шукати більш надійне рішення. Зважаючи на вразливі місця та недоліки WEP, Wi-Fi Alliance створив стандарт Wi-Fi Protected Access (WPA), який є підмножиною 802.11i (2003).

**Протокол WPA** був задуманий не як принципово інше рішення, а як покращення недоліків функції безпеки WEP. Так, ідея мала сенс, бо покращена технологія була придатна для роботи з існуючими продуктами Wi-Fi, у яких увімкнено WEP. Технологія WPA переважно має три вдосконалення порівняно з WEP:

- посилення шифрування даних через використання протоколу цілісності тимчасового ключа (TKIP) за допомогою алгоритму хешування та додавання функції перевірки цілісності. TKIP – це функція хешування тимчасового ключа, яка є альтернативою WEP і не потребує нового обладнання. Як і WEP, TKIP використовує потоковий шифр RC4, як для шифрування, так і для дешифрування, і всі залучені сторони мають використовувати один секретний ключ;
- автентифікація користувача, яка відсутня у WEP, через розширений протокол автентифікації (EAP). Для WEP доступ до бездротової мережі був на основі спеціальної апаратної MAC-адреси комп'ютера, яку порівняно легко виявити та викрасти. EAP побудовано на більш безпечній системі шифрування з відкритим ключем, щоб забезпечити доступ до мережі лише авторизованим користувачам мережі [8];
- новий спосіб цілісності повідомлення (MIC) для TKIP. Для усунення слабкості алгоритм планування ключа було запропоновано замінити на хеш-алгоритм альтернативного тимчасового ключа (ATKH), який був названий Michael [9].

**Протокол WPA2.** Покращення безпеки Wi-Fi здавалось очевидним, однак через короткий час, у 2004 році було випущено протокол WPA2, який справді підвищив рівень захищеності мереж Wi-Fi. З 2006 року WPA офіційно замінено на WPA2. Однією з найбільш значних змін між WPA та WPA2 є обов'язкове використання алгоритмів AES і введення CCMP (Counter Cipher Mode with Block Chaining Message Authentication Code Protocol) як заміна TKIP. Проте TKIP досі зберігся в WPA2 як резервна система, а також для взаємодії з WPA. Ця оновлена версія WPA ґрунтується на механізмі мережі високої безпеки (RSN) і працює у двох режимах:

- персональний режим, або спільний ключ Pre-Shared Key (WPA2-PSK), використовує спільний пароль доступу та зазвичай застосовується в домашніх мережах;
- корпоративний режим (WPA2-EAP) орієнтується на мережі організацій і комерційне використання.

В обох режимах використовується протокол CCMP, що ґрунтується на алгоритмі розширеного стандарту шифрування (AES), який забезпечує автентифікацію та цілісність повідомлення. Протокол CCMP є більш надійним, ніж протокол TKIP, що використовується в WPA, тому його використання ускладнює атаки зловмисників.

Відповідно до режиму Personal, автентифікація досягається за допомогою Pre-Shared Key та в режимі Enterprise завдяки стандарту IEEE 802.1X/EAP.

Характерною рисою протоколу WPA2 є також технологія WPS (Wi-Fi Protected Setup), яка спрощує підключення WPA2 у житлових мережах, тобто для підключення в домашньому середовищі можливо використовувати звичайні PIN-коди або шляхом натискання кнопок на роутері. Такий функціонал сильно сприяв поширенню WPA2, але водночас став джерелом вразливостей, адже призводить до можливості здійснення атак на стандарт WPS, тому для кращої безпеки рекомендовано вимкнути опцію WPS і, якщо можливо, прошивку. Цей протокол підтримується всіма можливими сертифікованими пристроями Wi-Fi. Наразі переважна більшість локальних бездротових мереж тривалий час працює в режимі захисту протоколу WPA2.

При цьому вразливості вказаного протоколу вже добре відомі [10, 11]. За час використання WPA2 було виявлено немалий список вразливостей:

Технологія «рукостискання» Four-way handshaking (FWH), переустановлення ключів парного шифрування PTK-TK, ключ групи GTK, ключ цілісності IGTK; переустановлення ключа STK. використання шифрування TKIP, механізм Fast Transition (FT), тунельне підключення TDLS та його одноразовий ключ TPK, технологія керування фреймами, відкриті контрольовані фрейми, дисоціація відносно точок доступу, повторна автентифікація.

Перелічені вразливості визначаються певними рівнями, які залежать від режимів і налаштувань мережі та можуть призводити до значного набору атак, щодо мережі Wi-Fi [12]:

- атаки за словником;
- офлайн-атаки підбору паролю за перехопленим рукостисканням;
- KRACK (Key Reinstallation Attacks), основою якого є 4-етапне рукостискання;
- зламування WPS;
- Advanced Stealth Man-in-the-middle (Hole 196);
- атака на PSK;
- Brute force;
- Evil Twin.

Тривалий досвід використання WPA2 призвів у 2018 року альянс Wi-Fi до представлення якісно нового протоколу WPA3, який мав би знівелювати перелічені вразливості та сильно знизити ефективність атак.

**Протокол безпеки WPA3**, який на сьогодні є найсучаснішим щодо надання покращеного захисту бездротових мереж, визначається новими опціями шифрування й автентифікації. Головною особливістю WPA3 є те, що забороняється використовувати застарілі протоколи безпеки, можлива інтеграція лише з протоколом WPA2, але за визначених умов використання.

Провідні характеристики WPA3:

1) строге використання стійкості мережі з використанням захищених кадрів керування PMF;

2) підтримка й оновлення тенденції режимів безпеки Enterprise та Personal;

3) ініціювання автентифікації SAE;

4) сприяння перехідного режиму безпеки, яке визначається підтримкою взаємодії з пристроями, які послуговуються протоколом WPA2;

5) обмін ключами Dragonfly за криптографічним експоненціальним методом SPEKE;

6) технологія Opportunistic Wireless Encryption (OWE), яка дає змогу шифрувати з'єднання між пристроєм користувача та точками доступу мережі за механізмом Діффі – Хеллмана, що, зі свого боку, сприяє генеруванню різних ключів дешифрування для кожного з користувачів. Примітка: технологія OWE також шифрує відкриті мережі;

7) додаткова опція DPP дає змогу здійснювати автентифікацію клієнтів за допомогою тегів NFC та QR-коду.

**Покращені нововведення в протоколі WPA3 порівняно з протоколом WPA2:**

1) алгоритм SAE замінює WPA-PSK;

2) шифрування OWE замінює відкриту автентифікацію 802.11;

3) протокол DPP замінює технологію WPS.

Таблиця 1

Порівняльні показники протоколів безпеки

| **Складові механізму протоколу** | **WEP** | **WPA** | **WPA2** | **WPA3** |
|---|---|---|---|---|
| Механізм шифруванння | RC4 (Використання – IV вектора) | RC4/TKIP | AES/CCMP CCMP/TKIP | GCMP-256 Simultaneous Authentication of Equals (SAE) |
| Розмір ключа шифрування | 40 біт | 128 біт | 128 біт | 256 біт |
| Шифрування ключа пакета | Конкатенація | Змішаний | Немає потреби | Немає потреби |
| Шифрування ключа управління | Жодного | 802.1x | 802.1x | WLAN 802.11i OWE |
| Шифрування зміни ключа | Жодного | Для кожного пакета | Немає потреби | Немає потреби |

Продовження таблиці 1

| Складові механізму протоколу | WEP | WPA | WPA2 | WPA3 |
|---|---|---|---|---|
| Апаратна сумісність | Можливе розгортання на поточній апаратній інфраструктурі | Можливе розгортання як на поточному, так і на старішому обладнанні | Старіші карти мережевого інтерфейсу не підтримуються | Можливе розгортання на обладнанні WPA2 з оновленнями ПЗ |
| Управління ключами | Жодного | 4-стороннє рукостискання | 4-стороннє рукостискання | 384-розрядна еліптична крива Діффі – Хеллмена (ECDH) |
| Автентифікація | Слабка | 802.1x – EAP | 802.1x – EAP | Одночасна автентифікація (SAE) |
| Цілісність даних | CRC-32 | MIC (Michael) | CCM | 256-bit Broadcast/ Multicast Integrity Protocol |
| Атака з найбільшою загрозою | Вразливий до зламу ключа методом проб і помилок фрагментації Bittu та DoS-атак на відмову в обслуговуванні, включно з різноманітними DoS-атаками | Вразливий до зламу ключа методом проб і помилок, Ohigashi-Morii, WPA-PSK і Dos-атаки | Вразливий до DoS-атак через незахищені контрольні кадри та підробку MAC-адрес | Вразливий до видалення SSL, Evil Twin Attack і DNS Spoofing |

WEP відрізняється від WPA і WPA2 як необхідністю специфічного обладнання, так і використовуваним програмним забезпеченням. У таблиці 1 наведено відмінності між WEP, WPA, WPA2 і WAP3 з точки зору підходів до шифрування й автентифікації. У WPA2 атака «грубою силою», імовірно, використовувалася для виявлення пароля Wi-Fi. Це пояснюється процедурою передбачення пароля, яка може відбуватися в автономному режимі. У WPA3 атак грубої сили успішно уникають, оскільки нові стандартні дозволи не дають можливості хакерам передбачати пароль в автономному режимі. Це і є перевагою у використанні WPA3-SAE порівняно з WPA2-PSK. Відмінності в процесі шифрування досліджуваних стандартів безпеки Wi-Fi залежать від використовуваних алгоритмів шифрування, які певним чином впливають на рівень захищеності мережі.

На рис. 2 показано як приклад вибір політики безпеки мережі Wi-Fi з доступних на маршрутизаторах, а саме на застарілому обладнанні, після 2006 року:

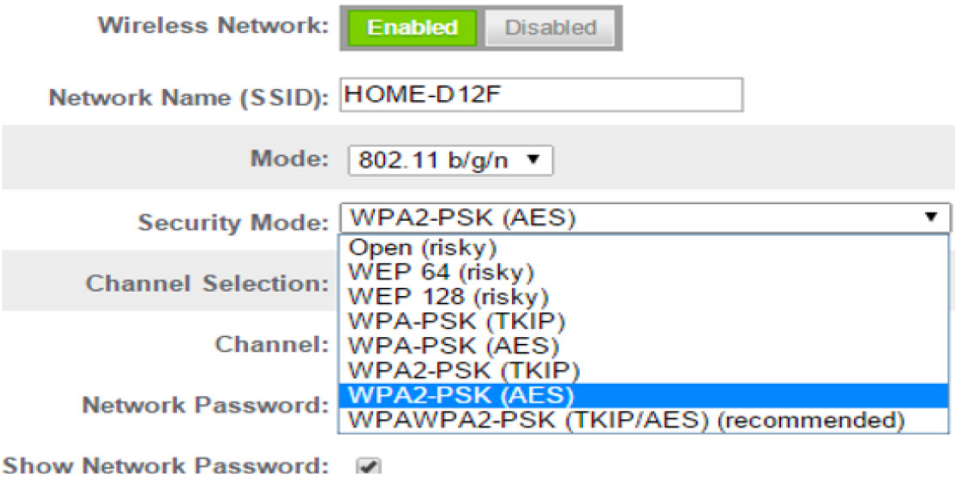


Рис. 2. Налаштування захисту локальної бездротової мережі

У наведеному прикладі налаштування режиму безпеки обладнання не містить можливості використання WPA3. З точки зору стандарту WPA3, поточне обладнання є застарілим. Однак це не означає, що воно встановлено до 2018 року. Виробники продовжують випускати наведене обладнання, тобто воно має попит, хоча поряд із ним саме на ринку маршрутизаторів достатньо номенклатури, яка забез-

печує функціонування WPA3, але підтримує також WPA2. Це свідчить, що протокол WPA3 за ці всі роки з моменту появи у 2018 році не став домінуючим. До того ж цей протокол як засіб захисту до цього часу займає малу частку мереж Wi-Fi. Так, за даними публічної бази даних вебсайту **WiGLE** [13], онлайн-ресурсу, який збирає інформацію про різні бездротові точки доступу Wi-Fi по всьому світу, кількість мереж Wi-Fi перевищувала 1 мільярд екземплярів уже у 2021 році і продовжувала зростати. У таблиці 2 наведено дані WiGLE про частки протоколів безпеки в мережах Wi-Fi за останні три роки.

Таблиця 2

Частки протоколів безпеки в мережах Wi-Fi

| **Протокол безпеки Wi-Fi (шифрування)** | **Період часу** | | |
|---|---|---|---|
| | **01.01.2021 – 31.12.2021** | **01.01.2022 – 31.12.2022** | **01.01.2023 – 03.05.2023** |
| WPA3 | 0,003% | 0,05% | 0,195% |
| WPA2 | 69,98% | 71,93% | 72,86% |
| WPA | 4,45% | 3,9% | 3,58% |
| WEP | 4,6% | 4,0% | 3,7% |
| Незашифровані мережі | 2,9% | 2,5% | 2,4% |
| Невідомі мережі | 18,1% | 17,6% | 17,4% |

Як ми бачимо, існує тенденція до зростання частки WPA3 та зниження часток застарілих протоколів. Але привертає увагу дивовижно низький рівень частки WPA3. Причини цього явища ще достеменно не досліджені, проте існує ряд факторів, які стримують темпи поширення цього протоколу. Цей протокол практично усунув проблеми атаки KRACK (Key Reinstallation Attacks), які найбільше загрожують протоколу WPA2. Справа в тому, що ці атаки небезпечні тим, що використовують вразливості в самому стандарті Wi-Fi, а не в окремих продуктах чи рішеннях. Тобто проблема полягає в тому, що цей протокол теж має свої вразливості. У таблиці 1 вони відображені в загальному вигляді. Ряд дослідників ґрунтовно дослідили вразливості WPA3 [14–17]. Далі розглянемо одну з найбільш загрозливих. Ряд провідних експертів, незважаючи на значну кількість проведених покращень для зниження вразливостей, які притаманні протоколу WPA2, не визначають принципово нового протоколу WPA3. Натомість це сертифікація, що визначає, які ж існуючі протоколи має підтримувати пристрій. Дещо спрощено WPA3 вимагає підтримки рукостискання Dragonfly і його єдиною новою функцією є перехідний режим, де WPA2 і WPA3 одночасно підтримуються для зворотної сумісності. І сама безпека протоколу WPA3 та його рукостискання Dragonfly – під сумнівом. На рис. 3 показано схему обміну, між клієнтом і точкою доступу відома під назвою Dragonfly.

Варіант рукостискання Dragonfly, який використовується в WPA3, також відомий як одночасна автентифікація рівних (SAE). У 2020 році М. Вангоф і Е. Ронен показали, що має місце витік, пов'язаний із паролем, що дає можливість здійснити словникову атаку в автономному режимі.

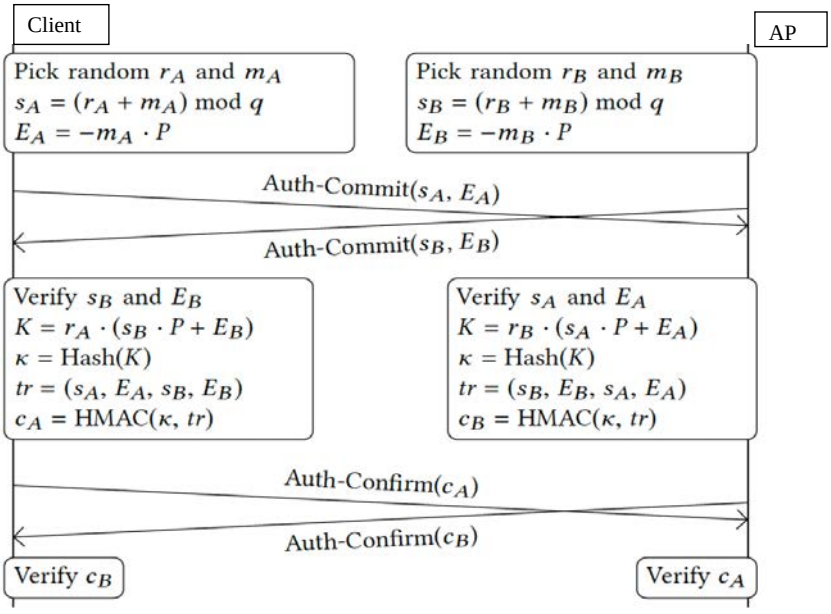


Рис. 3. Рукостискання Dragonfly WPA3 [15]

Ця атака отримала образну назву Dragonblood унаслідок метафори графічного зображення 4-етапного рукостискання. Відкриття атаки Dragonblood стало неприємною несподіванкою для Wi-Fi Alliance, який щойно отримав найбільше оновлення за 14 років. Однак постачальники обладнання продовжують впровадження WPA3, керуючись тим, що KRACK WPA2 є більш серйозною загрозою, оскільки стосується самого стандарту, тоді як Dragonblood переважно використовує недоліки реалізації, пов'язані з витоками побічних каналів. Альянс Wi-Fi опублікував деякі вказівки щодо реалізації, яких мають дотримуватися виробники, щоб забезпечити безпечну зворотну сумісність реалізацій WPA3. Незважаючи на це, впровадження WPA3 йде дуже повільно. Наразі швидка глобальна заміна протоколу WPA2 сучасною технологією WPA3 практично неможлива, і такий стан ще порівняно довгий час зберігатиметься.

Проведений аналіз відтворює, що сучасні бездротові мережі, зокрема в міському середовищі, являють собою конгломерат підмереж, кожна з яких підтримується різними протоколами й технологіями інформаційного захисту. При цьому жоден протокол не гарантує стовідсоткового захисту, хоча експерти оцінюють їх можливості по-різному – від відкритого доступу до найвищого, у ролі якого виступає WPA3. Постають питання: як оцінити рівень захищеності мереж Wi-Fi загалом? як конкретному споживачу вибрати прийнятний варіант захисту своєї мережі?

Ці питання виникають у користувачів Wi-Fi скрізь, але особливо вони актуальні в міському середовищі, де спостерігається щільна забудова. У міському середовищі Wi-Fi мереж багато, а щільність забудови дає змогу легко організувати «підслуховування». Проте спочатку треба встановити реальний стан оснащення протоколами захисту мереж Wi-Fi для середовища певного населеного пункту. У ролі об'єкта в цьому дослідженні було вибрано середовище міста Києва.

**Методологія дослідження**

Для визначення оснащення протоколами захисту мереж Wi-Fi у міському середовищі треба мати відповідні методології. Такими є методології Wardriving або Access Point mapping. У нашому дослідженні ми використали Wardriving. Термін Wardriving описує техніку, що використовується для пошуку й відображення сигналів WLAN у певному місті чи районі. Wardriving зазвичай визначається як переміщення транспортним засобом, зазвичай автомобілем, у визначеній географічній зоні й сканування для перерахування точок бездротового доступу та їх робочий стан у режимі реального часу. Використання Wardriving дає змогу отримати реальні дані про оснащення протоколами безпеки цих типів мереж. Було проведено пасивне дослідження зі збору лише загальнодоступних даних, які передаються з кожної бездротової точки. Будь-які спроби проникнення в будь-яку мережу не допускались. За допомогою використаної техніки Wardriving [18], яка базується на застосуванні мережевої карти, GPS-приймачів, відбувається процес «непомітного» сканування безпровідних мереж на виявлення розташувань і налаштування найближчих мереж із використанням авто як засобу пересування. Вибрана техніка дає змогу проводити легальне сканування бездротових мереж, оскільки законами не заборонено збирати дані з бездротових мереж, як і жодні закони не забороняють переписувати номери будинків у місті. Wardriving не є хакерською, незаконною або шкідливою діяльністю для перевірених бездротових мережевих пристроїв або їхніх власників. Назва цієї методології походить від кінофільму Wargames і не має нічого спільного з будь-якими військовими застосуваннями. Як методологія дослідження міських мереж Wi-Fi, технологія Wardriving використовується досить давно. Перше автоматизоване дослідження безпеки бездротової мережі було проведено Шиплі наприкінці 1999 року в Каліфорнії та вперше оприлюднене у 2001 році [18]. З того часу дослідники з різних країн, застосовуючи Wardriving, провели численні дослідження стану захищеності міських мереж, причому на цьому матеріалі можна дослідити як еволюцію бездротових в містах [19–24] різних країн і континентів, так і еволюцію їх захисту. В Україні було досить давно проведено одне таке дослідження, а саме у 2008 році [25], результати якого мають дуже суттєву різницю із сучасними даними.

**Проведення сканування мереж**

Для збору даних було вибрано стільниковий пристрій із параметрами: display=SP1A.210812.016.G780GXXU3CVI4, model=SM-G780G, release=12, device=r8q, board=kona, brand=samsung, Було встановлено програмне забезпечення для сканування мережі застосунок WiGLE в середовищі ОС Android.

З врахуванням величезних масштабів мегаполіса Києва повне дослідження міста вимагало надто багато часу й ресурсів, тому були проведені вибіркові заміри на території м. Києва для сканування бездротової мережі на правому та лівому берегах річки Дніпро, загальною площею близько 9 км$^2$, у травні 2023 року. Дослідженням вибірково охоплені спальні райони, промислові зони та певна частка центру міста. З огляду на стан безпеки, тобто травень 2023 року, конкретна локація сканувань не вказується. За результатами сканування у вибраних ділянках всього було виявлено 331 мережу типу Wi-Fi. З них 61 було з невизначеним (невідомим) SSID.

Сканування відбувалось автоматично під час пересування та використання застосунку WiGLE у визначених місцях із налаштуванням параметрів для сканування та фільтрування.

Параметри для сканування охоплювали: основні налаштування щодо часу сканування Wi-Fi (50 мс), коли швидкість близько 8 км/год (без зупинки), перевірка призупинення Wi-Fi (1,5 хв); GPS-на-

лаштування щодо тайм-ауту мережевого місцезнаходження (1 хв), час між GPS-оновленнями (1 с) та таймаутом (15 с). Параметри для фільтрування такі: відображення відкритих мереж, WEP та WPA-типізованих мереж.

Дані, зібрані унаслідок сканування мереж у визначених місцевостях міста Києва з використанням застосунку WiGLE, були експортовані у формат .csv для утворення датасету. Дані не завантажувалися на ресурс сайту WiGLE, а зберігалися локально для подальшого аналізу даних.

Дані у форматі csv були завантажені в локальне сховище Jupyter Notebook для подальшої обробки датасету.

Результат обробки даних на основі датафрейму, який відображає поточну ситуацію розподілу використання протоколів безпеки у досліджених ділянках міста Києва мереж Wi-Fi, наведено на рис. 4.

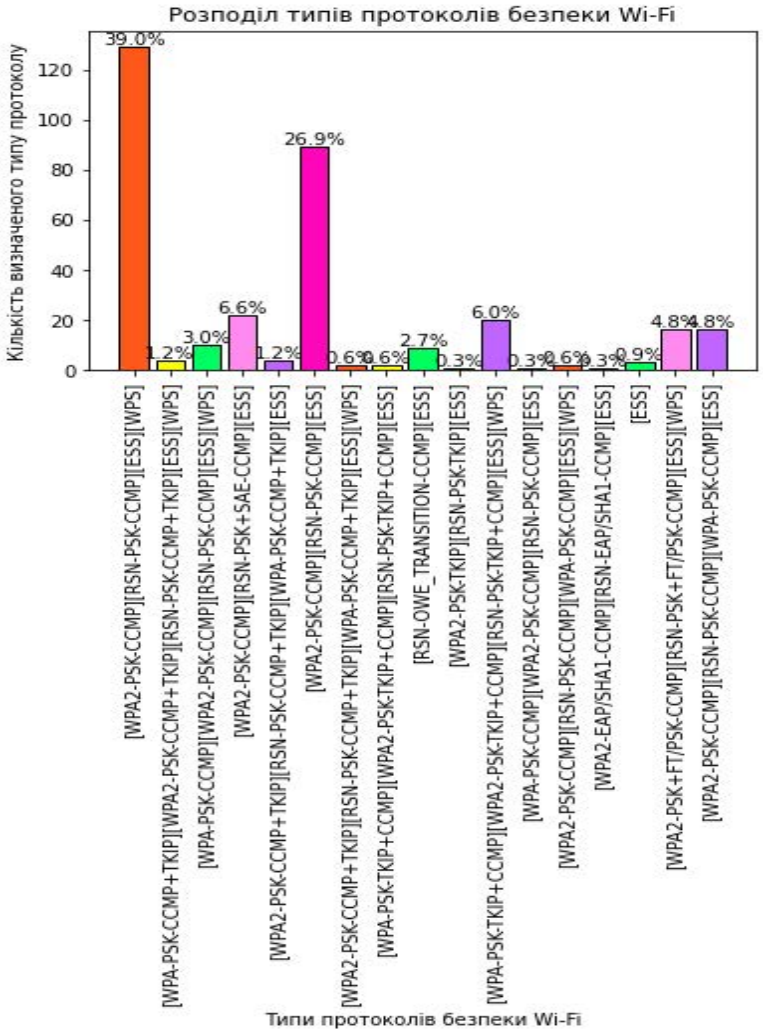


Рис. 4. Розподіл типів протоколів захисту мереж Wi-Fi у мережах м. Києва

**Оцінювання рівня безпеки**

Як можна бачити з представленої діаграми, розподіл систем кіберзахисту мереж Wi-Fi за даними вибірки для м. Києва дає досить строкату картину, яка, власне, дає можливість отримати тільки номінальну шкалу вимірювань. Перше, на що звертаємо увагу, – це відсутність застарілого протоколу WEP. Як ми бачили, за даними світової статистики, цей протокол продовжує залишатися в ужитку, хоча його частка вже досить мала, але в м. Києві (за результатами отриманої вибірки) він практично відсутній. Не виявлено також і використання новітнього протоколу WPA3. Переважає використання протоколу WPA2 різних версій, є певна частка протоколу WPA і невелика частка публічних мереж Wi-Fi як з певним захистом, так і повністю відкритих. Однак як оцінити загалом стан і цілісність захищеності мереж у мегаполісі на прикладі Києва? Для відповіді на це питання треба провести два вартісних і трудомістких дослідження:

- протестувати на зламування всі варіанти протоколів з усіма відомими для них атаками;
- просканувати всі мережі мегаполіса.

Малоймовірно, що коли-небудь ці завдання будуть виконані. Доводиться скористатися вибірковими даними, а рівень захищеності версій протоколу оцінювати альтернативним способом, опираючись на відносні оцінки рівня захищеності типу протоколу, яке надає обладнання мереж Wi-Fi.

Однак для оцінювання рівня захищеності цього недостатньо. У таблиці 3 наведено результати упорядкованого списку типів протоколів безпеки за рівнем захищеності разом з оцінкою захищеності за інтервальною шкалою. Слід зазначити, що йдеться не про абсолютну оцінку рівня захищеності, а про відносну. Тобто ту, яка відповідає поточному рівню можливостей технологій захисту бездротових мереж. Далі покажемо, як було отримано таблицю 3, у якій перша колонка позначає тип протоколу, друга – бренд Wi-Fi Alliance, третя – стандарт IEEE 802.11.

Побудуємо спочатку рангову шкалу вимірювань, тобто визначимо порядок, за якого певна політика безпеки, реалізована через компоненти технічної реалізації протоколів, буде краща чи гірша, ніж інша протиставлена їй політика. А потім побудуємо інтервальну шкалу.

Таблиця 3

Оцінка рівня захисту типів протоколу бездротових мереж

| **Сервіс захисту** | **Протокол Wi-Fi alians** | **Відповідність протоколу IEEE 802.11i-2004 (RSN)** | **Технології зручності** | **Частка використання в міському середовищі, %** | **Рівень захисту*** | **Індекс відносної безпеки** |
|---|---|---|---|---|---|---|
| P1 | WPA2-EAP/SHA1-CCMP | EAP/SHA1-CCMP | – | 0,3 | вс | 85 |
| P2 | WPA2-PSK-CCMP | PSK+SAE-CCMP | – | 6,6 | вс | 82 |
| P3 | WPA2-PSK-CCMP | PSK-CCMP | – | 26,9 | вс | 80 |
| P4 | WPA2-PSK-TKIP | PSK-CCMP | – | 0,3 | с | 75 |
| P5 | WPA2-PSK-CCMP | PSK-CCMP | WPS | 39 | с | 73 |
| P6 | WPA2-PSK-CCMP | PSK-CCMP | WPS, FT | 4,8 | с | 70 |
| P7 | [WPA2-PSK-CCMP] [WPA-PSK-CCMP] | PSK-CCMP | – | 4,8 | нс | 65 |
| P8 | [WPA2-PSK-CCMP] [WPA-PSK-CCMP] | PSK-CCMP | WPS | 0,6 | нс | 60 |
| P9 | [WPA2-PSK-CCMP+TKIP] [WPA-PSK-CCMP+TKIP] | PSK-CCMP+TKIP | – | 1,2 | нс | 58 |
| P10 | [WPA2-PSK-CCMP+TKIP] [WPA-PSK-CCMP+TKIP] | PSK-CCMP+TKIP | WPS | 0,6 | нс | 56 |
| P11 | [WPA-PSK-CCMP] [WPA2-PSK-CCMP] | PSK-CCMP | – | 0,3 | нс | 50 |
| P12 | [WPA-PSK-CCMP] [WPA2-PSK-CCMP] | PSK-CCMP | WPS | 3 | нс | 46 |
| P13 | [WPA-PSK-CCMP+TKIP] [WPA2-PSK-CCMP+TKIP] | PSK-CCMP+TKIP | WPS | 1,2 | нс | 46 |
| P14 | [WPA-PSK-CCMP+TKIP] [WPA2-PSK-CCMP+TKIP] | PSK-CCMP+TKIP | – | 0,6 | нс | 43 |
| P15 | [WPA-PSK-CCMP+TKIP] [WPA2-PSK-CCMP+TKIP] | PSK-CCMP+TKIP | WPS | 6 | нс | 43 |
| P16 | OWE_TRANSITION-CCMP | OWE_TRANSITIO-CCMP | – | 2,7 | нс | 40 |
| P17 | OPEN AUTENTIFICATION | OPEN 802.11 | – | 0,9 | нз | 10 |

*Примітка: * вс – вище за середній, с – середній, нс – нижче за середній, нз – незадовільно*

З погляду інформаційного захисту вибір протоколу з комбінації технологій визначає вибір політики безпеки. Як показує практика, користувачі бездротових мереж не завжди ставлять інформаційну безпеку на перше місце. Значну роль також грають цінові фактори, зручність встановлення, налагодження й експлуатація. Для цього розробники обладнання надають широкий спектр рішень. Зо-

крема, це стосується технології спрощеного підключення або безшовного роумінгу Wi-Fi. У принципі, це правильно, бо сприяє розширенню використання бездротових мереж. Проте ці рішення ще більш ускладнюють проблему захищеності мереж Wi-Fi, бо самі вони додають власні вразливості в загальну проблему захищеності.

Дослідники інформаційної безпеки бездротових мереж виділили декілька категорій, які визначають рівень захищеності [26].

**Автентичність**: автентичність вказує на відмінність авторизованих користувачів від неавторизованих користувачів. Будь-які підключені вузли в бездротових мережах мають спочатку пройти ідентифікацію щодо один одного за вже існуючими та вказаними даними й авторизацію щодо отримання відповідних прав доступу.

**Конфіденційність**: конфіденційність передбачає обмеження доступу до бездротової мережі неавторизованим користувачам і блокування доступу таких користувачів до бездротової мережі та розголошення інформації.

**Цілісність**: цілісність означає збереження інформації, її точність і надійність під час передачі через бездротову мережу.

**Наявність**: ця характеристика передбачає, що ця бездротова мережа дійсно доступна авторизованим користувачам для доступу за їхнім запитом.

Реалізація цих категорій залежить від показників безпеки, описаних у таблиці . Усі ці категорії технічного плану, які безпосередньо реалізують політику безпеки.

Крім того, ми вважаємо доцільним додати сюди категорію, яка безпосередньо не бере участі в реалізації варіанта технічного рішення, проте може вплинути на рівень захищеності бездротової мережі. Пропонуємо назвати цю категорію терміном «зручність».

**Зручність**: означає зручність встановлення, налаштування, використання. Наприклад, технологія WPS дає змогу зручно встановлювати обладнання в мережу, але разом із тим породжує нові вразливості.

Спочатку проведемо групове ранжування – виділимо групи типів протоколів за оцінками «добре», «посередньо», «нижче середнього», «незадовільно».

Оцінка «відмінно» тут відсутня, оскільки таку оцінку потенційно мав би протокол WPA3, але через виявлені вразливості його максимально можлива оцінка не 100, а 95.

Оцінки «добре» заслуговують протоколи P1–P3, оскільки до складу протоколу WPA2 входять сучасні засоби автентифікації, конфіденційності та цілісності.

Оцінки «посередньо» заслуговують протоколи P4–P6, оскільки до складу протоколу WPA2 входять дещо застарілі засоби автентифікації, конфіденційності та цілісності, або присутні засоби зручності.

Оцінки «нижче середнього» заслуговують протоколи P7–P15, оскільки до складу протоколу WPA2 входять різні варіанти переважно застарілих засобів автентифікації, конфіденційності та деякі типи протоколів, які мають засоби зручності. Але найважливіше, що впливає на таке оцінювання, – це гібридний характер цих типів протоколів, а саме можливість роботи в застарілому протоколі WPA. У цей інтервал потрапляє також і P16, хоча до класу гібридних не належить, а дає певні засоби захисту для користувачів відкритих мереж.

Нарешті оцінку «незадовільно» отримує протокол P17, призначений для обслуговування відкритих мереж без засобів захисту.

Щоб отримати інтервальні оцінки, потрібно провести ранжування типів протоколів у межах виділених класів і надати їм оцінку за певною шкалою. Ці результати представлені в таблиці 3. Причому типи протоколів ранжуються за критеріями більш надійних технологій захисту, наявності або відсутності технологій зручності.

Для визначення числової оцінки візьмемо 100-бальну шкалу. Для цієї вибірки максимальна можлива оцінка дорівнює 85, для оцінки «посередньо» максимальна можлива оцінка 75, для оцінки «нижче середнього» максимальна можлива оцінка 60, а мінімальна – 40. Те, що нижче за 40, оцінюється як «незадовільно». Числові значення оцінок типів протоколів у межах виділеної шкали отримані шляхом опитування групи спеціалістів з наступним усередненням.

Далі на підставі інтервальних оцінок захищеності типів протоколів і визначених їх часток в бездротових мережах міського середовища Києва, можемо вирахувати відносну оцінку захищеності мереж загалом.

$$RISN=\sum_1^n Pi * Fi, \tag{1}$$

де *RISN* – відносний індекс захищеності мереж міського середовища;
*Pi* – і-й протокол з таблиці 3;
*Fi* – частка і-го протоколу.

Підставляючи значення відповідних колонок Таблиці 3, у формулу отримаємо:

$$RISN = 70{,}113.$$

Тобто згідно з даними вибірки сканованих бездротових мереж у середовищі м. Києва загальна відносна оцінка їх захищеності лежить в діапазоні «посередньо».

**Висновки**

Дослідження оцінки рівня захищеності бездротових мереж у міському середовищі потребувало аналізу механізмів захисту протоколів, які мають забезпечувати надійне шифрування, автентифікацію та цілісність з'єднань. Для встановлення реального стану бездротових мереж проведено сканування бездротових мереж із використанням методології Wardrive у вибіркових локаціях міста Києва. Отримані дані після обробки були використані для оцінювання рівня захищеності мереж Wi-Fi для м. Києва на основі розробленої методики. Згідно з отриманою оцінкою рівень захищеності бездротових мереж міського середовища перебуває в задовільному стані.

Проте в перспективі отримане значення можна покращувати впровадженням нового протоколу безпеки WPA3, хоча це буде потребувати витрат. Однак існує можливість уже зараз як покращити загальний рівень захищеності мереж, так і підвищити індивідуальну безпеку, зокрема, завдяки кращому налаштуванню рівня безпеки обладнання.

***Література***

1. Kohlios C., Hayajneh T. A. Comprehensive Attack Flow Model and Security Analysis for Wi-Fi and WPA3. Electronics 7 (11), 284, 2018, pp. 1–28.

2. W-F Alliance, "Value of wi-fi" 2022, URL: https://www.wi-fi.org/ discover-wi-fi/value-of-wi-fi (дата звернення: 10.11.2023).

3. Wireless Security and the IEEE 802.11 Standards GIAC Security Essentials Certification (GSEC) Practical Assignment Version 1.4c. URL: https://www.giac.org/paper/gsec /4214/wireless-security-ieee-80211-standards/106760 (дата звернення: 10.11.2023).

4. Edney Jon, Arbaugh William A. Real 802.11 Security: Wi-Fi Protected Access and 802.11i. Publisher: Addison Wesley, Pub Date: July 15, 2003, ISBN: 0-321-13620-9, 451 p.

5. ANSI/IEEE Std 802.11, 1999 Edition (R2003), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Section 8, Authentication and privacy, Sub section 8.2.1, p. 61.

6. Borisov N., Goldberg I., Wagner D. Intercepting Mobile Communications: The Insecurity of 802.11. URL: http://www.isaac.cs.berkeley.edu/isaac/ mobicom.pdf (дата звернення: 10.11.2023).

7. Fluhrer S., Mantin I., Shamir, A., Weaknesses in the Key Scheduling Algorithm of RC4. URL: http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf (дата звернення: 05.11.2023).

8. Stuart J. Kerry et al. Response from the IEEE 802.11 Chair on WEP Security, IEEE 802.11 Working Group. URL: http://slashdot.org/articles/01/02/15/1745204.shtml (дата звернення: 01.11.2023).

9. Housley R., Whiting D. *Temporal Key Hash*. IEEE 802.11-01/550r3, Dec 2001.

10. Srikanth V., Reddy I. Wireless security protocols (WEP, WPA, WPA2 & WPA3) [Електронний ресурс] // JETIR. – 2019. – URL: https://www.jetir.org/ papers/ JETIRDA06001.pdf.

11. The State of Wi-Fi® Security Wi-Fi CERTIFIED™ WPA2 TM Delivers Advanced Security to Homes, Enterprises and Mobile Devices Wi-Fi Alliance® – URL: https://davidhoglund. typepad.com/files/20120229_state_of_wi-fi_security_09may2012_updated_cert.pdf.

12. Caneill M. Attacks against the WiFi protocols WEP and WPA [Електронний ресурс] / M. Caneill, J. Gilis. – 2010. – Режим доступу до ресурсу: https://matthieu.io/dl/papers/wifi-attacks-wep-wpa.pdf.

13. [Електронний ресурс] : [Вебсайт]. – Електронні дані. –2023. – URL: https://wigle.net/ (дата звернення: 12.09.2023). – Назва з екрана.

14. Опірський І., Максимів Н., Женчур М. Дослідження безпеки стандарту Wi-Fi Protected Access 3 (WPA3*). Безпека інформації* 2023. Том 29. С. 21–31.

15. Vanhoef M., Ronen E. Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd 2020 // IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2020, pp. 517–533, doi: https://doi.org/10.1109/SP40000.2020.00031.

16. Bednarczyk Mariusz, Piotrowski Zbigniew Will WPA3 really provide Wi-Fi security at a higher level? // Proc. SPIE 11055, XII Conference on Reconnaissance and Electronic Warfare Systems, 27 March 2019; doi: https://doi.org/10.1117/12.2525020.

17. Braga D. De Almeida, Fouque P, Sabt M. Dragonblood is still leaking: Practical cache-based side-channel in the wild // ACSAC. ACM, 2020, pp. 291–303.

18. Hurley C., Rogers R., Baker B., Thornton F. WarDriving & Wireless Penetration Testing. Canada: Syngress, 2007. 446 p.

19. Yek S. Wily attackers seek wireless networks in Perth, Western Australia foreasy targets // European Conference on Computer Network Defence, London,United Kingdom, 2006, pp. 125–136.

20. Sarkar I., Abdullah A. H.Exploring wireless network security in Auckland City through warwalking field trials // Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT2011. January 2011.

21. Chih-Ta L., Hira S., Donald J. Wireless network security // Unitec New Zealand, 2004, pp. 337–339.

22. Mara´czi M. Wardriving in Eger // Proceedings of the IEEE 13th International Symposium on Applied Computational Intelligence and Informatics, IEEE, Timisoara, Romania, May 2019, pp. 127–130.

23. Valchanov H., Edikyan J., Aleksieva V. A study of wi-fi security in city environment//Proceedings of the IOP Conference Series: Materials Science and Engineering, IOP, October 2019.

24. Nasr E., Jalloul M., Bachalaany J., Maalouly R. Wi-fi network vulnerability analysis and risk assessment in Lebanon // Proceedings of the International Conference of Engineering Risk, Beirut, May 2019.

25. Дослідження захищеності Wi-Fi мереж в м. Києві [Електронний ресурс] : Електронні дані. – Київ, 2008. – URL: https://www.kaa.org.ua/22-articles/57-96.html (дата звернення: 12.11.2023). – Назва з екрана.

26. Yulong Jia Zhu, Xianbin Wang, Hanzo Lajos. A Survey on Wireless Security: *Technical Challenges. Recent Advances and Future Trends.* September 2016. Vol. 104. No. 9.

### *References*

1. Kohlios, C., Hayajneh, T. (2018). A. Comprehensive Attack Flow Model and Security Analysis for Wi-Fi and WPA3. *Electronics 7* (11), 284, pp. 1–28.

2. W-F Alliance, "Value of wi-fi" (2022). [Online]. Available: https://www.wi-fi.org/ discover-wi-fi/value-of-wi-fi.

3. Wireless Security and the IEEE 802.11 Standards GIAC Security Essentials Certification (GSEC) Practical Assignment Version 1.4c [Online]. Available: https://www.giac.org/paper/gsec/ 4214/wireless-security-ieee-80211-standards/106760.

4. Jon Edney, William A. Arbaugh Real 802.11 Security: Wi-Fi Protected Access and 802.11i. Publisher: Addison Wesley, Pub Date: July 15, 2003, ISBN: 0-321-13620-9, 451 p.

5. ANSI/IEEE Std 802.11, 1999 Edition (R2003), Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Section 8, Authentication and privacy, Sub section 8.2.1, 61 p.

6. Borisov, N., Goldberg, I., Wagner, D. Intercepting Mobile Communications: The Insecurity of 802.11 [Online]. Available: http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf.

7. Fluhrer, S., Mantin, I., Shamir, A., Weaknesses in the Key Scheduling Algorithm of RC4 [Online]. Available: http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf.

8. Stuart J. Kerry et al. Response from the IEEE 802.11 Chair on WEP Security, IEEE 802.11 Working Group [Online]. Available: http://slashdot.org/articles/01/02/15/1745204.shtml.

9. Housley, R., Whiting D. (2001). *Temporal Key Hash*. IEEE 802.11-01/550r3.

10. Srikanth, V., Reddy, I. (2019). *Wireless security protocols (WEP, WPA, WPA2 & WPA3)* / JETIR [Online]. Available: https://www.jetir.org/papers/JETIRDA06001.pdf.

11. The State of Wi-Fi® Security Wi-Fi CERTIFIED™ WPA2 TM Delivers Advanced Security o Homes, Enterprises and Mobile Devices Wi-Fi Alliance® January 2012.

12. Caneill, M., Gilis, J. (2010). Attacks against the WiFi protocols WEP and WPA [Electronic resource] / M. Caneill. Available: https://matthieu.io/dl/papers/wifi-attacks-wep-wpa.pdf.

13. [Electronic resource]: [Website] (2023). Electronic data [Online]. Available: https://wigle.net/ (Date: 12.09.2023).

14. Opirskyi, I., Maksymiv, N., Zhenchur, M. (2023). Doslidzhennia bezpeky standartu Wi-Fi Protected Access 3 ["Security research of Wi-Fi Protected Access 3 (WPA3) standard"]. *Bezpeka informatsii.* Vol. 29. № 1, pp. 21 [ in Ukrainian].

15. Vanhoef, M., Ronen, E. (2020). Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd. IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, pp. 517-533, doi: https://doi.org/ 10.1109/SP40000.2020.00031.

16. Mariusz Bednarczyk, Zbigniew Piotrowski (2019). Will WPA3 really provide Wi-Fi security at a higher level? // Proc. SPIE 11055, XII Conference on Reconnaissance and Electronic Warfare Systems, doi: https://doi.org/10.1117/12.2525020.

17. Braga, D. De Almeida, Fouque P, Sabt M. (2020). Dragonblood is still leaking: Practical cache-based side-channel in the wild in ACSAC. ACM, pp. 291–303.

18. Hurley, C., Rogers, R., Baker, B., Thornton, F. (2007). *WarDriving & Wireless Penetration Testing.* Canada: Syngress, 446 p.

19. Yek, S. (2006). Wily attackers seek wireless networks in Perth, Western Australia foreasy targets // European Conference on Computer Network Defence, London,United Kingdom, pp. 125–136.

20. Sarkar, I., Abdullah, A. (2011). Exploring wireless network security in Auckland City through warwalking field trials // Proceedings of the 13th International Conference on Advanced Communication Technology (ICACT2011), IEEE, Gangwon.

21. Chih-Ta, L., Hira, S., Donald, J. (2004). Wireless network security // Unitec New Zealand, pp. 337–339.

22. Mara´czi, M. (2019). Wardriving in Eger // Proceedings of the IEEE 13th International Symposium on Applied Computational Intelligence and Informatics, pp. 127–130, IEEE, Timisoara, Romania, May.

23. Valchanov, H., Edikyan, J., Aleksieva, V. (2019). A study of wi-fi security in city environment // Proceedings of the IOP Conference Series: Materials Science and Engineering, IOP.

24. Nasr, E., Jalloul, M., Bachalaany, J., Maalouly, R. (2019). Wi-fi network vulnerability analysis and risk assessment in Lebanon // Proceedings of the International Conference of Engineering Risk, Beirut.

25. Doslidzhennia zakhyshchenosti Wi-Fi merezh v m. Kyievi [ Study of the security of Wi-Fi networks in Kyiv] (2008). [Online]: [Website]. Avaible: https://www.kaa.org.ua/22-articles/57-96 (date of application: 12.11.2023) [in Ukrainian]

26. Yulong Jia Zhu, Xianbin Wang, Hanzo Lajos (2016). A Survey on Wireless Security: *Technical Challenges. Recent Advances and Future Trends*. Vol. 104. No. 9.

# КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

**УДК 004.9**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-02**

## ТЕОРЕТИЧНІ ОСНОВИ ФУНКЦІОНАЛЬНОГО ПРОЄКТУВАННЯ ШТУЧНОГО ОБ'ЄКТА Й ОРГАНІЗАЦІЇ ДІЯЛЬНОСТІ ДЛЯ ЙОГО КОМП'ЮТЕРНОЇ РЕАЛІЗАЦІЇ

***Антипенко Б. А.,*** *аспірант, кафедра інформаційних технологій, факультет електроніки та інформаційних технологій, Сумський державний університет, Суми, Україна*
*https://orcid.org/0000-0003-1350-8914, antip.bogdan8@gmail.com*

***Марченко А. В.,*** *канд. техн. наук, доцент, кафедра інформаційних технологій, факультет електроніки та інформаційних технологій, Сумський державний університет, Суми, Україна*
*https://orcid.org/0000-0003-2003-3531, a.marchenko@cs.sumdu.edu.ua*

***Неня В. Г.,*** *канд. техн. наук, доцент, кафедра інформаційних технологій, факультет електроніки та інформаційних технологій, Сумський державний університет, Суми, Україна*
*http://orcid.org/0000-0002-0965-9149, v.nenja@cs.sumdu.edu.ua*

***Антипенко В. П.,*** *канд. техн. наук, доцент, кафедра інформаційних технологій, факультет електроніки та інформаційних технологій, Сумський державний університет, Суми, Україна*
*https://orcid.org/0000-0001-8198-1848, v.antypenko@cs.sumdu.edu.ua*

***Анотація.*** *У статті представлено рекурсивний метод формування опису будь-якого штучного об'єкта, що забезпечує його практичну комп'ютерну підтримку, незалежно від типу об'єкта. Діяльність із розробки нових штучних об'єктів вважається виробничою діяльністю. Вона є предметом контрольно-управлінського, організаційно-розпорядчого регулювання. Така діяльність має три аспекти: проєктування, конструювання й естетичне оформлення. Останні два є досить простими щодо контролю й управління процесом розробки проєкту штучного об'єкта, оскільки базуються на раніше прийнятій і затвердженій концепції. Однак у першому випадку, коли проєктований об'єкт розглядається як такий, що містить багато складних компонентів і процесів, які в них відбуваються, вибирається підхід поетапної декомпозиції. Це процес розробки опису об'єкта алгоритмізовано на основі формального технічного завдання (ТЗ). Застосовується системний підхід, що означає вибір компонентів, які спеціально об'єднуються в один блок і забезпечують відповідність вимогам ТЗ. Опис об'єкта з використанням ICOM-підходу методу IDEF0 формує інтерфейси, потрібні для функціонування об'єкта. Проєктування завершується за умови відсутності ТЗ для уточнення опису компонента. ТЗ не складається після отримання елементарної задачі, для вирішення якої розробляється конструкція компонента, після вибору попередньо розробленого компонента та після вибору компонента стороннього виробника. Ці три випадки охоплюють усі можливі варіанти результатів декомпозиції функції та конструкції, яка її реалізує, і гарантують завершення організованого рекурсивного процесу деталізації опису проєктованого об'єкта. У висновку маємо розроблену методику проєктування штучного технічного об'єкта на основі покрокової декомпозиції цього процесу, яка забезпечує практичну комп'ютерну реалізацію останнього, незалежно від типу об'єкта.*

***Ключові слова:*** *рекурсивний метод, опис технічного об'єкта, штучний об'єкт, комп'ютерна підтримка, декомпозиція, опис компонентів, організація процесу проєктування.*

# THEORETICAL FOUNDATIONS OF FUNCTIONAL DESIGN OF AN ARTIFICIAL OBJECT AND ORGANIZATION OF ACTIVITIES FOR ITS COMPUTER REALIZATION

***Bohdan Antypenko,*** *Postgraduate Student at the Department of Information Technologies, Faculty of Electronics and Information Technologies, Sumy State University, Sumy, Ukraine https://orcid.org/0000-0003-1350-8914, antip.bogdan8@gmail.com*

***Anna Marchenko,*** *PhD in Engineering, Senior Lecturer at the Department of Information Technologies, Faculty of Electronics and Information Technologies, Sumy State University, Sumy, Ukraine https://orcid.org/0000-0003-2003-3531, a.marchenko@cs.sumdu.edu.ua*

***Viktor Nenia,*** *PhD in Engineering, Associate Professor, Department of Information Technologies, Faculty of Electronics and Information Technologies, Sumy State University, Sumy, Ukraine http://orcid.org/0000-0002-0965-9149, v.nenja@cs.sumdu.edu.ua*

***Viktoriia Antypenko,*** *PhD in Engineering, Associate Professor, Department of Information Technologies, Faculty of Electronics and Information Technologies, Sumy State University, Sumy, Ukraine https://orcid.org/0000-0001-8198-1848, v.antypenko@cs.sumdu.edu.ua*

***Abstract.*** *This article presents a recursive method of forming a description of any artificial object, which provides its practical computer support, regardless of the object type. The activity of developing new artificial objects is considered a production activity. It is the subject of control, management, organizational and administrative regulation. This activity has three aspects: designing, construction and aesthetic design. The last two are quite simple according to control and management of the development process of the artificial object project, as they are based on a previously accepted and approved concept. However, in the first case, when the designed object is considered as containing many complex components and their processes, a staged decomposition approach is chosen. This process of developing an object description is algorithmized on the basis of a formal Terms of Refences (ToR). A systematic approach is used. This means the selection of components which on purpose are combined into one unit and ensure compliance with the ToR requirements. Object description using the ICOM-approach of the IDEF0 method forms the interfaces necessary for the object functioning. The designing is completed if there is no ToR to clarify any component description. The ToR is not formed after obtaining the elementary task for which the component designing is developed, after selecting a previously developed component, and after selecting a component from a third-party manufacturer. These three cases cover all possible variants of the results of the decomposition of the function and the structure that implements it, and guarantee the completion of an organized recursive process of detailing the description of the designed object. In conclusion, the authors have developed a methodology for designing an artificial technical object of any which provides its practical computer implementation.*

***Key words:*** *recursive method, description of technical object, artificial object, computer support, decomposition, components description, organization of the designing process.*

**Вступ**

Закономірність людського суспільства про все зростаюче споживання товарів полягає в тому, що діяльність людей у побуті, на виробництві та в суспільній сфері вимагає нових видів об'єктів: більше якісних, більше функціональних, більше зручних, більше простих у застосуванні. При цьому бізнес, як рушійна сила нашого прогресу, докладає всіх зусиль для того, щоб такі об'єкти з'явилися на ринку якомога швидше. Багато в чому ці вимоги закладаються та частково реалізуються на етапі проєктування. Саме цей момент зумовлює вимогу до проєктування про швидку й надійну розробку проєкту майбутнього об'єкта з гарантуванням заданого рівня якості. У цих умовах проєктування потрібно розглядати як технологію та виробничий процес. При цьому попередньо наведені аспекти слід або врахувати, або забезпечити можливість проєктувальникам їх реалізовувати. Проєктування, як і інші виробничі процеси, потребує своєї інструментальної та машинної підтримки, а також автоматизації управління цим процесом. Бажання їх розробити керує дослідженнями авторів, окремі результати яких оприлюднюються для обговорення й удосконалення.

**Пов'язані роботи**

Важливість тематики з проєктування об'єктів зумовила значну кількість публікацій. Це визначає суб'єктивність як вибору робіт для аналізу, так і вибору точок зору на ті чи інші аспекти проєктування. Концептуально наша позиція полягає в тому, що проєктування є технологією, але в окремих моментах (за недостатнього розвитку теорії та технології – ми ж завжди проєктуємо щось нове) має характер ремесла (Craft). У цьому ми підтримуємо Suzanne Rivard [1]. Хоча процес становлення теорії проєктування йде паралельно з практикою проєктування, але запропонований у [1] спіралеподібний

процес формування теорії проєктування не може бути перенесений на процес створення проєкту об'єкта. Ітерації, а вони неминучі, повинні бути зведені до мінімуму, та носити локальний характер. При цьому не є припустимими глобальні цикли повторення всього процесу чи його окремих фаз або етапів. Усе повинно вирішуватися в межах однієї проєктної операції. Крім того, проєкт об'єкта, що створюється, постійно повинен бути в такому стані, який дає змогу виконувати контроль вимог до цього об'єкта.

Указані вимоги є можливими лише за умови організації процесу проєктування за підходом «згори донизу», оскільки це єдиний підхід, який дає змогу розглядати опис об'єкта як цілісного утворення. Методологія ця ґрунтовно опрацьована, завдяки чому можна достатньо обґрунтовано й наглядно представити результати проєктування [2]. Методологія Structured Analysis & Design Technique (SADT) отримала всебічну підтримку та подальший розвиток і набула статусу стандартів Integrated Computer-Aided Manufacturing (Integrated DEFinition IDEF). Розроблені свого часу ці стандарти й досі відіграють важливу роль, а головному стандарту функціонального моделювання IDEF0 немає альтернативи. Єдине, чого не достає, то це детальності, яка потрібна для формалізації моделі й отримання на цій основі конкретних значень показників, які дають змогу об'єктивно оцінювати призначення об'єкта. Центральна ідея методології SADT зосередження на головних аспектах модельованих об'єктів спрацювала і спрацьовує. Але вона залишила в тіні питання забезпечення конкретності виконуваного аналізу. До цього часу пропонується декілька способів трансляції компіляції діаграм для концептуального моделювання (IDEF0, UML тощо) в інші форми моделей та врешті в машинний код. Пов'язано це з необхідністю проведення синтаксичного й семантичного контролю моделі та можливістю об'єктивної перевірки ймовірних варіантів реалізації моделі. Так, у роботі [3] пропонується здійснювати перетворення діаграм діяльності UML, призначених для моделювання та верифікації бізнес-процесів організації, у YAWL моделі, які мають формальну семантику, засновану на мережах Петрі. Також використовуються кольорові та звичайні мережі Петрі [4–6]. Досягнуті певні успіхи, але автори всіх робіт із трансформації моделей вводять множини правил, обмежень і винятків. Разом із тим не забезпечується взаємно однозначна відповідність між моделями в різних формах, особливо семантичної. Підтвердженням того факту, що резерви методології SADT ще не вичерпалися, слугує робота [7], у якій ця методологія застосовується до функціонального моделювання технічних об'єктів хімічного виробництва.

**Метод дослідження**

Виконаний аналіз концептуального методу структурного аналізу та проєктування [1] і прикладів його застосування [2, 3, 7] показав його можливості й переваги. Він використовуватиметься як основний інструмент виконання проєктних процедур. Метод відомий і тому немає потреби його викладати.

Організація виконання проєктних процедур є питанням у загальному випадку проблемним. Це пов'язано з потребою в застосуванні комплексу засобів автоматизації проєктувальних робіт до проєктування різних штучних об'єктів (технічних, інформаційних тощо), які мають різний склад і різні схеми об'єднання свої складових в одне ціле, що має свої унікальні властивості та призначення для виконання оригінальних функцій. Це означає, що наперед неможливо скласти програмний код моніторингу, контролю та керування процесом проєктування. При цьому розробка методів проєктування та працездатних схем їх реалізації з урахуванням орієнтації на застосування комп'ютерів завжди впирається в питання «А чи буде алгоритм реалізовано? Чи буде кількість операцій скінченою?». Ці питання є фундаментальними, вони перебувають у центрі уваги вчених-математиків. Для пошуку відповідей на поставлені питання доцільно звернутися до теорії алгоритмів, а саме її розділу про рекурсивні функції [8]. Цей розділ написаний на сучасному формальному рівні математичної логіки.

У теорії алгоритмів [8] на основі робіт Чьорча, Поста, Тюрінга та Маркова стверджується, що задача є вирішуваною, якщо процес її рішення вдається звести до рекурсивного процесу, тобто скінченої множини однакових дій, які однаково повторюються. Це буде другим методом, який ми використовуватимемо.

Звичайно, використовується також системний аналіз, якому надається формалізована форма. Це дає змогу виконати алгоритмізацію і подальшу програмну реалізацію, що є необхідним для застосування системного аналізу в інформаційних технологіях проєктування.

У разі потреби під час викладення матеріалу будуть наведені або особливості використання методів, або їх трактування, або вдосконалення методів (у чому воно полягає).

**Виконання досліджень**

*Досліджувані процеси*. Поняття проєктування, з погляду онтології, характеризує два процеси, пов'язані зі штучними об'єктами. Далі розглядатимемо кожний із них.

По-перше, проєктування є одним з етапів життєвого циклу будь-якого штучного об'єкта. На цьому етапі останнього власне ще не існує. Об'єкт представлений проєктом. Це сукупність документів у паперовому та/або електронному вигляді. Наприклад, розрахунково-пояснювальні записки, файли моделей і даних, макети тощо. У проєкті наведено опис майбутнього об'єкта. У ньому зазначаються як основи, так і деталі функціонування в об'ємі, необхідному і достатньому для конкретного вироб-

ництва та використання або застосування. Водночас на попередньому рівні опису забезпечується можливість опису об'єкта чи будь якої його складової як єдиного цілого і на наступному за ним рівні як складеного, але таких, що виконують ту саму функцію, оскільки досліджуваний об'єкт розглядаємо і як ціле, і як складене. Він містить у собі певний набір компонентів зі своїми визначеними властивостями. Тобто сам проєкт містить матеріали з детальним описом особливостей функціонування як самого об'єкта, так і всіх його складових. Суть виконання цього етапу життєвого циклу полягає в наповненні проєктних документів відповідною інформацією. Повинні заноситися дані про різні аспекти, які характеризують об'єкт. Цей опис слугує вхідними даними для ініціалізації та виконання наступних етапів проєкту. А саме призначений для конструювання та виробництва об'єкта в умовах вибраного підприємства. Опис об'єкта орієнтований на конкретну технологію виробництва: конкретні машини, інструменти, методи обробки матеріалів та/або енергії і інформації. Зрозуміло, що поки повністю не буде прийняте проєктне рішення, наступні проєктні операції не будуть розпочаті. Однак слід розуміти, що коректне завершення проєктної операції передбачає не тільки оформлення проєктної документації, але й підготовку, погодження технічного завдання на ще не вирішені (а в нашому випадку не деталізовані) проєктні рішення.

По-друге, проєктування – це назва діяльності фахівців щодо створення вказаного опису створюваного об'єкта. Це суто організаційна частина справи: хто, що, коли робить. Зрозуміло, що це два зовсім різні процеси, тож і підходи до їх опису й реалізації повинні бути різні, хоча і взаємопов'язані. Однак спочатку слід розуміти, як розвивається проєкт об'єкта, який є необхідною складовою його життєвого циклу. А вже потім, відповідно до цього, організувати його наповнення. І тільки після попереднього здійснення зазначеної процедури призначати організацію роботи фахівців із реалізації методики формування опису штучного об'єкта. На жаль, ці процеси зовсім випали з поля зору дослідників. Крім того, їх розгляд не зустрічається в публікаціях. Це обґрунтовує потребу в їхньому дослідженні.

Діяльність із розробки нових штучних об'єктів розглядаємо як виробничу діяльність. Вона є предметом контрольно-управлінського, організаційно-розпорядчого регулювання. Така діяльність має три особливості. Це проєктування, конструювання й естетичне оформлення (дизайн або художнє конструювання). Останні два є досить простими щодо контролю й управління процесом розробки проєкту об'єкта. Однак у першому випадку потрібно використовувати моделювання та симуляцію. Варто розглядати декілька варіантів. Саме такі процеси є предметом розгляду авторів.

Світовий досвід проєктування, узагальнений в [9], показує, що найбільш доцільним є підхід до проєктування «згори донизу», який передбачає поступову деталізацію попередньо прийнятих проєктних рішень загального характеру. Цей підхід дає змогу здійснювати покрокову верифікацію запропонованих проєктних рішень та управління процесом проєктування.

Вибір статусу проєктування як виробничого процесу передбачає його раціоналізацію і прагматичність. Тому було вирішено його алгоритмізувати на основі використання формального технічного завдання так, як це запропоновано у [10]. Згідно з підходом до проєктування «згори донизу» на найвищому рівні перебуває ТЗ на створення об'єкта. Воно містить конкретний формалізований опис головної виконуваної об'єктом функції з конкретними даними, обмеженнями тощо. Тому наступним кроком відбувається покрокова декомпозиція виконуваної розглядуваним об'єктом функції. На кожну з уведених під час декомпозиції функцій складається нове ТЗ. Кожне ТЗ на всіх рівнях декомпозиції представлено окремим документом, який містить формалізований опис виконуваної функції. Процес декомпозиції функцій закінчується тоді, коли кожна його поточна складова функції найнижчого рівня має файл спеціального ТЗ на розробку компонента конструкції, який функцію реалізує.

Наприклад, у вигляді елементарної задачі зі всією відомою необхідною інформацією, готового конкретного конструктивного рішення, яке попередньо реалізоване в організації або покупного компонента. Також усі вищезазначені технічні завдання містять дані про зв'язки всіх функцій між собою.

Наявність чіткого конкретного набору даних ТЗ забезпечує формування обґрунтованої команди керування на виконання й забезпечення умов для відповідальності за завершення у визначений строк (до призначеного терміну) роботи на основі виданого завдання. При цьому стає доступною можливість здійснення об'єктивного контролю. Наявність формалізованої форми ТЗ дає змогу налагодити моніторинг покрокового «нарощування» інформаційного опису проєктованого об'єкта у проєктних документах та оперативний контроль машинним способом за ходом виконання робіт і повнотою реалізації вимог до об'єкта. На всі паралельно виконувані додаткові функції формується своє власне ТЗ, і всі вони опрацьовуються паралельно.

Згідно з наведеною аргументацією було прийнято рішення щодо організації декомпозиції опису об'єкта, а також алгоритмізації діяльності команди проєктувальників (саме процес розробки проєкту об'єкта на основі формалізованого технічного завдання).

Для забезпечення однорідної (однотипової, з одним порядком виконання робіт за тими самими правилами) організації виконання процесу розробки проєкту об'єкта на початковому кроці створюється ТЗ на проєктування об'єкта для конкретного (заданого) виробництва**.** Цей процес є першим у реалізації запропонованого підходу. Усі наступні кроки реалізують таке:

1) за наявності будь-якого ТЗ планується та видається виробниче завдання на його реалізацію;
2) виконується процес декомпозиції функції;
3) для всіх прийнятих функцій виконується перевірка наявності готового проєктного рішення;
3а) за наявності готового проєктного рішення складається ТЗ на його перевірку і внесення до проєкту або на відповідне доопрацювання;
3б) за відсутності готового проєктного рішення перевіряється можливість купівлі компонента для реалізації функції;
3б1) за відсутності такої можливості складається ТЗ на декомпозицію функції;
3б2) за наявності виконується порівняння умов купівлі відповідного компонента та власної реалізації рішення та складаються відповідні ТЗ.

Зрозуміло, що задача функціонального проєктування є складною (складеною і комплексною) і тому потребує декомпозиції. Тому кожна задача повинна розбиватися на менші. Мається на увазі, що для кожної з них також потрібно створити своє ТЗ на проєктування кожного компонента об'єкта. Обов'язково кожна задача повинна містити дані про зв'язок із задачами свого рівня, а також вищим і нижчим.

Графічно декомпозиція процесу розробки проєкту об'єкта, а отже, виконуваних ним функцій може бути інтерпретована як деревовидні структури. Матрична форма опису таких структур як графів не накладає обмежень на кількість рівнів деталізації функцій об'єкта та структури процесів. Їх кількість залежить як від особливостей функціонування, будови та характеристик проєктованого об'єкта, так і від наповнення бази знань проєктної організації чи відповідного підрозділу.

Є три види задач, які розглядаються в цій роботі. Насамперед усі складні завдання, які не мають готових компонентів рішень, повинні підлягати подальшій деталізації. Тобто під час останньої будуть створені для виконання задачі щодо розробки відповідних ТЗ на проєктування необхідних компонентів. Також можуть бути використані вже готові проєктні рішення, наявні в базах знань організації. Тому деревовидна структура фіксує результати використання вже реалізованих у попередніх проєктах ТЗ. Тобто розробка останніх у цьому випадку непотрібна. Можливо, застосовується налаштування, підгонка, адаптація тощо. І нарешті, у деревовидній структурі можуть мати місце описи проєктних рішень із використанням об'єктів сторонніх розробників і шаблони завдань на закупівлю готових компонентів іззовні. Тобто будуть наявними на виконання задачі, які характеризуються своєрідними ТЗ на придбання необхідних і доступних на ринку компонентів (деталей, механізмів, інструментів, матеріалів, програмних бібліотек тощо). Декомпозиція процесу розробки проєкту об'єкта вважатиметься завершеною, коли на відповідному нижньому рівні буде до виконання або елементарна задача проєктування простої функції однієї зі складових цього об'єкта, яка не потребує подальшої деталізації, або на застосування вже готового рішення організації-розробника / проєктанта, або на закупівлю необхідного компонента.

Після свого завершення ця деревовидна структура відображатиме ієрархічно повний набір взаємозалежних задач щодо проєктування об'єкта як цілого, так і його окремих компонентів. Вони можуть містити завдання як на розробку ТЗ щодо проєктування певних складових цього об'єкта з унікальними властивостями, так і на застосування наявних проєктних рішень організації та на придбання готових компонентів.

У кінцевому підсумку така формалізація процесу розробки проєкту об'єкта сприятиме скороченню часу на планування змісту, порядку й об'єму робіт, у тому числі й у часі, завдяки автоматизації цього етапу життєвого циклу об'єкта.

Отже, встановлено сутність проєктування штучного об'єкта у двох аспектах: як процесу наповнення опису об'єкта і як діяльності фахівців із здійснення такого процесу. Запропоновано підхід до реалізації багатоаспектного процесу проєктування, який дає змогу контрольовано реалізувати поставлене завдання з дотриманням усіх вимог.

*Опис об'єкта як цілого*. Початок формування концептуальної частини проєкту. На першому кроці об'єкт проєктування розглядається як ціле, яке буде використовуватися у призначеному для цього середовищі. При цьому конкретизуються зв'язки об'єкта з іншими об'єктами. Опис цих зв'язків (інтерфейсу) виконується шляхом формування множин параметрів субстанцій та артефактів відповідно до прийнятих домовленостей у методології. До загальноприйнятих у методології SADT та IDEF0 позначень на діаграмах Input, Command, Output and Mehanism додано опис множин параметрів зовнішніх і внутрішніх, як це було виконано авторами в роботі [10]. Крім традиційного змістовного опису субстанцій та артефактів на діаграмах, було введено їх характеристики.

*Опис об'єкта як складеного*. Продовження формування концептуальної частини проєкту. Тепер виконується декомпозиція функції *F*. Це може бути як функція самого об'єкта, так і будь-якого його компонента. Формалізований опис процесу декомпозиції вищезазначеної функції *F* продемонстровано авторами в роботі [11].

*Організація рекурсивного процесу проєктування*. Будемо дотримуватися вибраного підходу, і завдання кожної роботи підтримуватиметься наявністю формального ТЗ на проєктування. Для першого кроку маємо ТЗ на проєктування об'єкта як цілого. Процес проєктування на початковій стадії по-

лягає в декомпозиції функцій. Виконуючи декомпозицію, слід брати до уваги накопичений своєю організацією досвід і використовувати такі функції, які вже практично реалізовані та допускають пряме застосування або використання після налаштування чи регулювання. У цьому випадку формується технічне завдання на організацію закупівлі необхідного компонента або на внесення до проєкту відповідних компонентів з іншого проєкту. Другий варіант передбачає використання такої функції, яка не потребує подальшої декомпозиції. У цьому випадку формується технічне завдання на конструктивне оформлення такої функціональності. Після повної реалізації цих двох варіантів технічні завдання не створюються, а останній функції присвоюється статус «Реалізовано». У разі потреби подальшої декомпозиції (це третій і останній варіант) формується технічне завдання за схемою, представленою авторами в роботі [11].

Оскільки з кожним рівнем декомпозиції функції стають усе простішими і простішими, врешті-решт для кожної останньої у дереві декомпозиції знайдеться або готове конструктивне рішення, або навіть готовий об'єкт. Це є гарантією завершення рекурсивного процесу, оскільки нові технічні завдання вже не генеруватимуться.

*Планування робіт на розробку проєкту об'єкта*. Планування робіт із проєктування штучного об'єкта представлено покроковою декомпозицією процесу розробки його проєкту та відображено на рисунку 1.

У разі отримання організацією проєкту до виконання між замовником і представником підприємства узгоджується і затверджується ТЗ (рис. 1, (1)) на виготовлення певного штучного об'єкта. Затверджене ТЗ (1) є вхідними даними внутрішньої предметної інформаційної підсистеми (ВПІПс), яка веде проєкт штучного об'єкта і є компонентом зовнішньої системи управління діяльністю проєктної організації, яка веде роботу проєктантів. Під час цієї дії відбувається ініціалізація початку першого етапу процесу проєктування штучного об'єкта – процесу розробки проєкту штучного об'єкта. На цьому етапі відбувається старт процесу створення проєктних документів з описом майбутнього штучного об'єкта. Проєктні документи мають бути наповнені певною інформацією щодо різних аспектів, які характеризують такий об'єкт. Мається на увазі, що вони повинні містити дані про деталі функціонування проєктованого об'єкта як цілого, так і його компонентів.

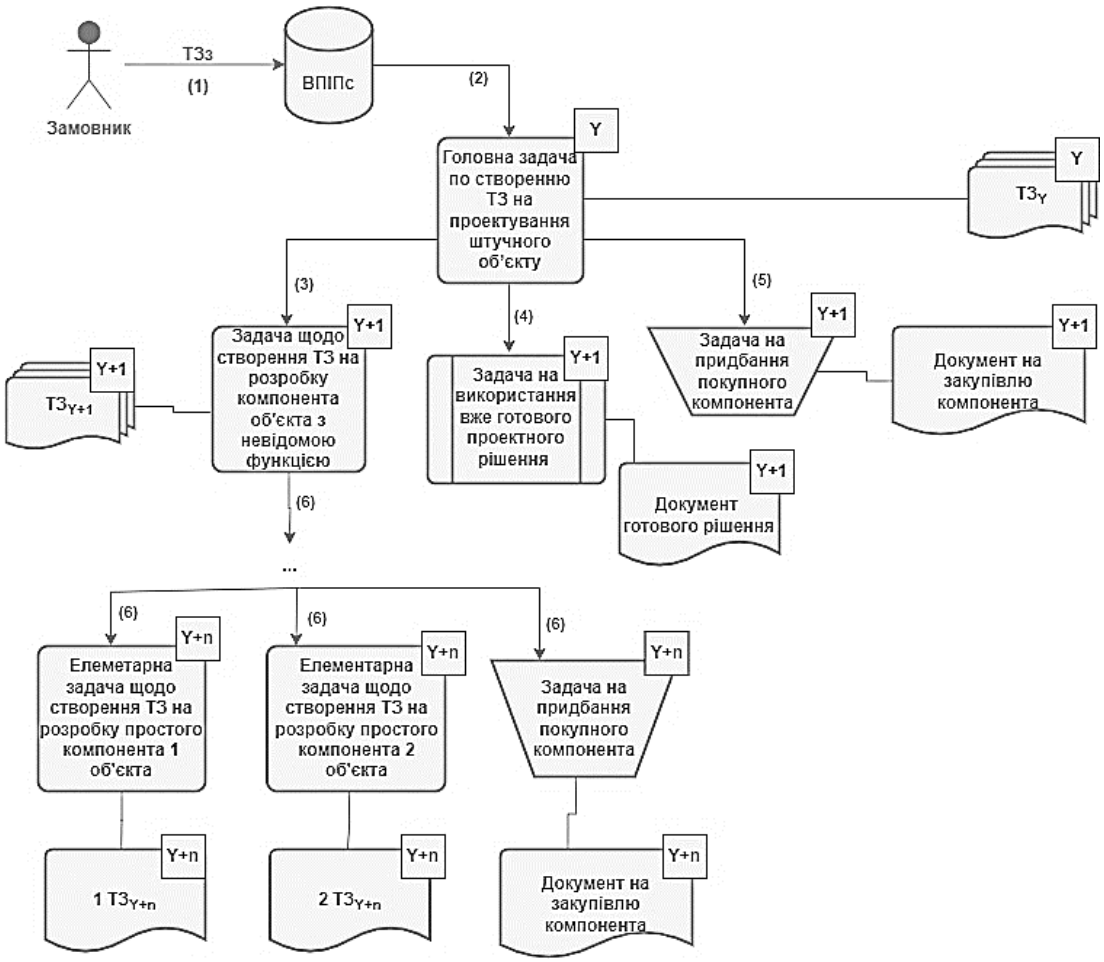


Рис. 1. Декомпозиція процесу проєктування штучного об'єкта

Далі розглянемо покрокову декомпозицію процесу розробки проєкту цього штучного об'єкта. Будемо виконувати планування робіт із проєктування останнього, використовуючи підхід «згори донизу», і представимо результати у вигляді деревовидної структури. На найвищому рівні Y відповідним фахівцем цієї організації розміщено головну задачу щодо створення ТЗ на проєктування цього штучного об'єкта як цілого (рис. 1, (2)) на основі ТЗ замовника із ВПІПс. Воно міститиме формалізований опис головної виконуваної ним функції з відомими вхідними та вихідними даними, обмеженнями, механізмами реалізації тощо, але без зазначення деталей алгоритму роботи цього штучного об'єкта. Ця головна задача є складеною і тому потребує розбиття на більш прості задачі для належної організації виконання процесу розробки проєкту представленого штучного об'єкта.

Отже, переходимо до формування наступного (дочірнього) рівня декомпозиції *Y + 1*. Вважаємо проєктований штучний об'єкт таким, що містить компоненти з невідомими функціями, компоненти у вигляді готових проєктних рішень у рамках цієї організації, і існуючі (наприклад, покупні) компоненти. У такому випадку для останньої формується задача зі своєрідним ТЗ на його придбання (рис. 1, (5)) і занесення до проєкту даних про інтерфейси. Для другого компонента створюється задача на використання вже готового проєктного рішення (рис. 1, (4)) за умови відсутності необхідності розробки ТЗ на його проєктування. Готове проєктне рішення є наявним завдяки раніше набутому досвіду організації під час реалізації попередніх проєктів. А для першого компонента буде сформована задача щодо створення ТЗ на його проєктування. ТЗ на проєктування компонента міститиме формалізований опис виконуваної ним функції з відомими вхідними та вихідними даними, обмеженнями, механізмами реалізації тощо, але знову без зазначення деталей алгоритму його роботи. Це так, оскільки вищезазначена задача також є складеною. А отже, вона й далі потребує подальшої декомпозиції. Цей процес розбиття складеної задачі на простіші повторюватиметься аналогічним чином (рис. 1, (6)) із додаванням до існуючої деревовидної структури наступних рівнів деталізації *Y + 2*, *Y + 3*, …, *Y + n*. Процес декомпозиції завершиться тоді, коли його найнижчий рівень *Y + n* буде представлений або елементарною задачею щодо створення ТЗ на проєктування відповідної простої складової цього штучного технічного об'єкта, яка не потребує подальшого розбиття, або задачею на застосування вже готового рішення, або задачею на закупівлю прийнятого як проєктне рішення певного компонента.

У підсумку маємо декомпозицію процесу розробки проєкту цього штучного технічного об'єкта у вигляді деревовидної структури. Така структура ієрархічно відображає певний набір взаємозалежних задач щодо створення відповідних документів з описом технічного об'єкта як цілого, так і його окремих компонентів. Ці проєктні документи також містять деталі про виконувані ними функції. Вищезазначені задачі можуть містити завдання як на розробку ТЗ на проєктування певних складових цього технічного об'єкта з унікальними властивостями, так і на застосування наявних готових рішень підприємства та/або на придбання компонентів ззовні.

Виділимо основні особливості організації робіт із формування життєвого циклу штучного об'єкта. Розбиття задач щодо проєктування об'єкта відбувається згори донизу за таким алгоритмом: формується задача найвищого рівня *Y* щодо створення ТЗ на проєктування об'єкта і створюються задачі рівня *Y + 1* (дочірні) щодо розробки відповідних проєктних документів зі всією належною інформацією. У разі потреби для певних задач рівня *Y + 1* формуються власні підзадачі за аналогічним алгоритмом, і так відбувається для кожного рівня деталізації, доки найнижчий рівень декомпозиції *Y + n* не буде представлений або елементарною задачею, яка не потребує подальшого розбиття, або задачею на застосування вже готового рішення організації, або задачею на закупівлю певного компонента.

Завершення задач проєктування технічного об'єкта відбувається знизу вгору таким чином: задача рівня *Y* вважається виконаною в тому випадку, якщо створено певний проєктний документ з описом відповідного компонента об'єкта й завершено всі її дочірні задачі рівня *Y + 1*. Тобто результатом виконання кожної з них є також створені певні проєктні документи з описом відповідних ним компонентів об'єкта. Набір готових проєктних документів рівня *Y + 1* становить результат виконання задачі рівня *Y*.

Подальше конструювання та виробництво штучного об'єкта відбувається за підходом «знизу вгору». Проєктування об'єкта є покроковим процесом, який розвивається в напрямі поступового спрощення поточно вирішуваних проєктних задач.

*Організація робіт із проєктування*. Ініціалізація робіт із проєктування починається після виявлення появи ТЗ на проєктування зі статусом «Не опрацьовується». Послідовно будуть виконані етапи планування, реалізації, контролю [11].

Введення механізму формалізованих технічних проєктних завдань дає можливість автоматизованого формування та здійснення контролю виконання виробничих завдань з їх реалізації. Таким чином, розроблено рекурсивний метод формування опису довільного штучного об'єкта та процесу його формування, що забезпечує їх практичну комп'ютерну реалізацію, незалежно від типу об'єкта.

**Висновки**

Розроблено методику проєктування штучного технічного об'єкта на основі покрокової декомпозиції цього процесу. При цьому виконано таке:

1. Встановлено сутність проєктування штучного об'єкта. Проаналізовано два аспекти: процес наповнення опису об'єкта та діяльність фахівців із здійснення такого процесу.

2. Запропоновано узагальнену методику опису функцій об'єктів проєктування як цілісних об'єктів. Це забезпечує необхідні дані для оцінювання варіантів і прийняття проєктних рішень і змістовного наповнення проєкту.

3. Запропонована узагальнена методика опису функцій об'єктів як складених дає змогу верифікувати проєктні рішення попередніх рівнів деталізації. Крім того, це дає необхідні дані не тільки для оцінювання варіантів і прийняття проєктних рішень, а й для обґрунтування їх із врахуванням причинно-наслідкових зв'язків між процесами, які відбуваються у складових об'єкта проєктування. Додатково підтримується однорідність змістовного наповнення проєкту й організації та планування процесу проєктування.

4. Запропоноване однорідне подання даних забезпечує рекурсивну організацію контрольованого процесу проєктування, що гарантує результативне його завершення або обґрунтоване доведення неможливості цього із фіксацією невирішених задач, що дає змогу обґрунтовано формувати плани дослідних робіт.

5. Проєктування об'єкта організовано як покроковий процес, який розвивається в напрямі поступового спрощення окремо вирішуваних проєктних задач.

6. На основі механізму формалізованих технічних проєктних завдань сформована можливість автоматизованого формування та здійснення контролю виконання виробничих завдань із їх реалізації.

Розроблено рекурсивний метод формування опису довільного штучного технічного об'єкта та процесу його формування. Однорідність усіх розроблених концептуальних моделей гарантує алгоритмізацію, що забезпечує їх практичну комп'ютерну реалізацію, незалежно від типу об'єкта. Запропонований підхід до реалізації процесу проєктування дає змогу контрольовано реалізувати поставлену задачу.

### *Література*

1. Rivard S. Theory Building: Neither an Art nor a Science, But a Craft. Advancing Information Systems Theories: Rationale and Processes, 2021. P. 131–159.

2. DeMarco T. Structured Analysis and System Specification. In: Broy, M., Denert, E. (eds) Software Pioneers. Springer, Berlin, Heidelberg, 2002.

3. Belghiat A., Oukhaf D. & Chaoui A. Transforming UML Diagrams to YAWL Models for Business Processes Analysis. In International Symposium on Modelling and Implementation of Complex Systems, Cham: Springer International Publishing, 2020. P. 279–293.

4. Nguyen Thanh T., Le Thanh N. & Hoang Thi Thanh H. Formalization of Business Processes and Business Rules Model using Colored Petri Nets. In The 5th International Conference on Future Networks & Distributed Systems, 2021. P. 42–47.

5. Fang K. & Lin S. An integrated approach for modeling ontology-based task knowledge on an incident command system. Sustainability, 11 (12), 2019. P. 3484.

6. Nguyen Thanh T., Le Thanh N., Hoang Thi Thanh H. & Ha Thi, T. VeBPRu: A Toolchain for Formally Verifying Business Processes and Business Rules. In Proceedings of the 2023 8th International Conference on Intelligent Information Technology, 2023. P. 15–19.

7. Moshev E., Meshalkin V. & Romashkin M. Development of models and algorithms for intellectual support of life cycle of chemical production equipment. Cyber-Physical Systems: Advances in Design & Modelling, 2020. P. 153–165.

8. Dean W. Recursive Functions. Stanf8ord Encyclopedia of Philosophy, 2021.

9. Cloutier R.J. & Hutchison N. Guide to the Systems Engineering Body of Knowledge (SEBoK), version 2.2. INCOSE, Stevens Institute of Technology, 2020.

10. Antypenko V., Nenia V., Marchenko A., Antypenko B., Kovpak A. Information technology for implementation the functional modeling of a technical object. Lecture Notes in Networks and Systems, Springer, Cham, 2021. P. 504–512.

11. Antypenko V., Nenia V., Marchenko A., Antypenko B. Recursive Method of Forming a Technical Object Description and Design Process Organization. Lecture Notes in Networks and Systems, Springer, Cham, 2022. P. 522–530.

### *References*

1. Rivard, S. (2021). Theory Building: Neither an Art nor a Science, But a Craft. *Advancing Information Systems Theories: Rationale and Processes*, P. 131–159.

2. DeMarco, T. (2002). Structured Analysis and System Specification. In: *Broy, M., Denert, E. (eds) Software Pioneers. Springer, Berlin, Heidelberg.*

3. Belghiat, A., Oukhaf, D., & Chaoui, A. (2020). Transforming UML Diagrams to YAWL Models for Business Processes Analysis. In *International Symposium on Modelling and Implementation of Complex Systems*, *Cham*: *Springer International Publishing*, P. 279–293.

4. Nguyen Thanh, T., Le Thanh, N., & Hoang Thi Thanh, H. (2021). Formalization of Business Processes and Business Rules Model using Colored Petri Nets. In *The 5th International Conference on Future Networks & Distributed Systems*, P. 42–47.

5. Fang, K., & Lin, S. (2019). An integrated approach for modeling ontology-based task knowledge on an incident command system. *Sustainability*, 11 (12), P. 3484.

6. Nguyen Thanh, T., Le Thanh, N., Hoang Thi Thanh, H., & Ha Thi, T. (2023). VeBPRu: A Toolchain for Formally Verifying Business Processes and Business Rules. In *Proceedings of the 2023 8th International Conference on Intelligent Information Technology*, P. 15–19.

7. Moshev, E., Meshalkin, V. & Romashkin, M. (2020). Development of models and algorithms for intellectual support of life cycle of chemical production equipment. *Cyber-Physical Systems*: *Advances in Design & Modelling*, P. 153–165.

8. Dean W. (2021). Recursive Functions. *Stanf8ord Encyclopedia of Philosophy*.

9. Cloutier, R.J. & Hutchison N. (2020). Guide to the Systems Engineering Body of Knowledge (SEBoK), version 2.2. *INCOSE*, *Stevens Institute of Technology*.

10. Antypenko, V., Nenia, V., Marchenko, A., Antypenko, B., Kovpak, A. (2021). Information technology for implementation the functional modeling of a technical object. *Lecture Notes in Networks and Systems*, *Springer*, *Cham*, P. 504–512.

11. Antypenko, V., Nenia, V., Marchenko, A., Antypenko, B. (2022). Recursive Method of Forming a Technical Object Description and Design Process Organization. *Lecture Notes in Networks and Systems*, *Springer*, *Cham*, P. 522–530.

**УДК 004.94**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-03**

# АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗБОРУ Й АНАЛІЗУ ДАНИХ ПРО ВАРТІСТЬ ЗЕМЛІ СІЛЬСЬКОГОСПОДАРСЬКОГО ПРИЗНАЧЕННЯ З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ

***Бутенко М. Є.,*** *Відкритий міжнародний університет розвитку людини «Україна», Київ, Україна*
*https://orcid.org/0009-0009-9545-6572, maksym.butenko.study@gmail.com*

***Павленко В. І.,*** *канд. фіз.-мат. наук, доцент,*
*Відкритий міжнародний університет розвитку людини «Україна», Київ, Україна*
*https://orcid.org/0000-0002-3958-0415, pavlenko.v@i.ua*

***Анотація.*** *У роботі розглянуто загальний підхід до архітектури програмного забезпечення, збору й аналізу даних за допомогою методів машинного навчання для передбачення вартості землі сільськогосподарського призначення. Наведено нові результати дослідження з використання методів LightGBM, Fast Tree, Fast Forest. Виділені етапи: збір даних, зберігання, дослідження та трансформація, агрегування, оптимізація та використання методів машинного навчання. Запропоновано використання декількох програмних застосунків для імплементації перерахованих етапів. Наведена архітектура взаємодії таких застосунків. Оглянуто використання даних про інфляцію та їх вплив на якість передбачення. До даних, зібраних на майданчиках з аукціонів, сайтів оголошень (перше джерело даних), додані дані із сайту Державної служби України з питань геодезії, картографії та кадастру (друге джерело даних). Набір даних, зібраний з аукціонів і оголошень, становить 2 123 рядки. З даних, наявних на сайті Державної служби України з питань геодезії, картографії та кадастру, відібрано інформацію про 102 924 ділянок. З першого джерела виділено повний набір усіх даних D1; набори даних D2 – множина D1 без даних про оренду та D3 – множина D1, де врахована інфляція за даними Національного банку України. З другого джерела виділено набір D4 – ділянки з типом «договір купівлі-продажу» за 2022–2023 роки; D5 – підмножина D4 для ділянок, вартість за 1 га яких не менше ніж 10 000 грн та не більше ніж 150 000 грн; D6 – множина D5, до якої додана інформація про відстані від великих міст України, та D7 – множина D6 з урахуванням інфляції. Використано методи машинного навчання для наборів даних, зібраних із цих джерел. Передбачення вартості з похибкою не більше за 10% вважається коректним. Результатом якості роботи машинного навчання залежно від алгоритму та набору даних є значення коефіцієнта детермінації до 0,897 та з відсотком коректних передбачень до 91. Для другого джерела точність оцінки гірша, ніж для першого, через меншу кількість атрибутів і відсутність інформації про оренду, термін оренди, родючість ґрунтів, деякі неточності у даних. Дослідження показують, що покращення якості початкового набору даних (збільшення повноти, репрезентативності) значно впливає на коефіцієнт детермінації та відсоток коректних передбачень навченої моделі.*

***Ключові слова:*** *машинне навчання, передбачення вартості землі, прогнозне моделювання, методи LightGBM, Fast Tree, Fast Forest.*

# SOFTWARE ARCHITECTURE FOR THE COLLECTION AND ANALYSIS OF AGRICULTURAL LAND PRICE DATA USING MACHINE LEARNING

***Maksym Butenko,*** *Open International University of Human Development "Ukraine", Kyiv, Ukraine*
*https://orcid.org/0009-0009-9545-6572, maksym.butenko.study@gmail.com*

***Volodymyr Pavlenko,*** *Ph.D., Ass. Prof. Open International University of Human Development "Ukraine", Kyiv, Ukraine*
*https://orcid.org/0000-0002-3958-0415, pavlenko.v@i.ua*

***Abstract.*** *The work presents a general approach to the software architecture for data collection and analysis using machine learning methods to predict the price of agricultural land. A brief review of recently published literature is made. New research results using LightGBM, Fast Tree, and Fast Forest methods to predict land price are provided. The highlighted stages include data collection, storage, exploration, and transformation, aggregation, optimization, and the use of machine learning methods. It is proposed to use several software applications for the implementation of these stages. An architecture diagram of the interaction of such applications is provided. The use of inflation data and the impact of such data on forecast quality reviewed. In addition to the dataset collected from the auctions and classifieds sites (first data source), data available on the website of the State Service of Ukraine for Geodesy, Cartography,*

*and Cadastre (second data source) were used. The dataset collected from first data source consist of 2123 rows, collected from seconds data source contains 815529 (further filtered to less rows due to different types of contract, year, etc). List of datasets for which machine learning was performed has been modified: attribute about soil type, distances from big cities, inflation were added. Information from the data available on the website of the State Service of Ukraine for Geodesy, Cartography, and Cadastre was converted to dataset, and it was further filtered and utilized, ranging from 75843 to 102924 rows. From the initial data set (D1), two additional ones were created: without data on land rent (D2) and with prices that take into account inflation (according to the National Bank of Ukraine) (D3). Experiments on removing the value of the regulatory monetary assessment of land reviewed. Also, for data source data was converted to four datasets. Dataset D4 – constructed using date filters for 2022 and 2023 for the "sale-purchase agreement" type. Rows without specified prices will be removed from the dataset. The total number of rows in dataset D4 is 102924. Dataset D5 – additionally apply a filter on the price per hectare of land, ranging from no less than 10000 UAH to no more than 150000 UAH. After this filtration, only 82241 parcels will remain in the dataset. Dataset D6 – expand dataset D5 with information about distances from the largest cities in Ukraine to the land. Dataset D7 – further expand dataset D6 with information about inflation. Thus, machine learning methods LightGBM, Fast Tree, and Fast Forest have been used for datasets collected from two different sources. If the cost prediction has an error of no more than 10%, it is considered correct. The result of the machine learning, depending on the algorithm and data set, for* $R^2$ *up to 0.897 and percentage of correct predictions up to 91%. Research shows that as the number of rows in the initial dataset increases, the quality of prediction improves. As for second data source – it have slightly worser prediction results then first data source, but second data source usage have shown how fast dataset can be extended with new parameters (e.g. soil type, distances from cities, inflation) to significantly increase prediction quality. Another possible way to improve the quality of the forecast could be the integration of certain data sets on non-market factors affecting land value (e.g., a map of potential contamination by explosive objects). The summary suggests further steps to improve the quality of predictions.*

***Key words:*** *machine learning, land price prediction, prediction modeling, LightGBM, Fast Tree, Fast Forest methods.*

**Вступ**

У статті [1] показано важливість використання методів машинного навчання для передбачення вартості земельних ділянок сільськогосподарського призначення в Україні. Особливо це актуально у зв'язку з важливими змінами в законах України щодо умов купівлі-продажу земельних ділянок. У цій статті пропонується загальний підхід до архітектури програмного забезпечення зі збору й аналізу даних методами машинного навчання з урахуванням впливу інфляції на передбачення вартості. Додатково використані дані Державної служби України з питань геодезії, картографії та кадастру «Інформація про здійснені транзакції по земельним ділянкам за період» [2].

Коректне передбачення визначається за умовою [3]:

$$K = \left| \frac{x - x_i}{x_i} \right| \le 0{,}1 , \tag{1}$$

де K – передбачення, x – передбачене значення, $x_i$ – реальне значення.

Відсоток коректних передбачень визначається за формулою

$$P = \frac{K_k}{K_a} * 100\% , \tag{2}$$

де $P$ – відсоток коректних передбачень, $K_k$ – кількість коректних передбачень, $K_a$ – кількість передбачень усього.

Результатом машинного навчання в роботі [1] отримано моделі оцінки вартості земельних ділянок, що мають коефіцієнт детермінації 0,818 і дають 77% коректних передбачень для тестового набору даних (92% для тренувального набору даних).

У роботі [4] розглянуто питання оцінки вартості землі в Республіці Шрі-Ланка за допомогою методів машинного навчання: Multiple Linear Regression, Random Forest, Support Vector Regression, Extra Tree Regression, XGBoost. На відміну від робіт [3, 5] (ці роботи були детально розглянуті в [1]), у цій роботі не вираховується певний відсоток коректних передбачень, але наводяться результати роботи алгоритмів у метриках MSE (середньоквадратична похибка), RMSE (коренева середньоквадратична похибка), MAPE (середня абсолютна похибка передбачення), $R^2$ (коефіцієнт детермінації), де $R^2$ досягає значень від 0,87 до 0,9 при MSE від 0.09 до 0.12 та RMSE від 0.31 до 0.35. Найкращі результати показало використання методу Random Forest та XGBoost.

У роботі [6] обговорюється передбачення вартості побудови «зелених будинків» за допомогою методів машинного навчання, і описані підходи можуть бути корисні для оцінки вартості земельних ділянок. Стаття детально описує загальний підхід до вирішення такої задачі, а саме які дані та яким

чином треба збирати, наведено методологію передбачення вартості за допомогою блок-схеми. Методологія подібна до запропонованої у [7] та в цій статті. У роботі порівнюється використання методів машинного навчання XGBoost, Random Forest, Deep Neural Network (DNN). Найкращим результатом роботи є модель передбачення з коефіцієнтом детермінації $R^2$ на рівні 0,96. Результат використання методу машинного навчання XGBoost був кращим за результат використання DNN ($R^2$ = 0,96 для XGBoost та $R^2$ = 0,95 для DNN). Автори доходять висновку, що використання методів машинного навчання дає змогу вирішувати поставлену задачу.

У роботі [8] описано підхід до використання методів Gradient boosting, G-RNN, Random Forest, SVR, ARIMA до передбачення вартості продукції сільськогосподарської діяльності. Як і для ринку продажу земельних ділянок, існують певні проблеми збору даних. Автори описують загальний підхід до збору даних та їх приведення до вигляду, необхідного для машинного навчання. Через наявність історичних даних за певний проміжок часу використано алгоритми аналізу часових рядів. У висновках зазначається, що проблема такого аналізу є складною та потребує якісних початкових даних, але й отримані результати можуть бути додатковим чинником для вирішення питань про плани на продаж продукції для господарств.

**Джерела даних та архітектура програмного забезпечення**

Розробка архітектури програмного забезпечення для навчання за допомогою методів машинного навчання є важливою для забезпечення ефективності, надійності та масштабованості аналізу даних. Використання простих алгоритмів без адекватної архітектури може призвести до ряду проблем, включно з низькою точністю моделей, складністю збору даних та їх зберігання, повільною обробкою великих обсягів даних, важкістю підтримки й розширення функціонала програмного забезпечення з плином часу. Розробка архітектури дає змогу структурувати процеси збору, обробки й аналізу даних, забезпечуючи оптимальне використання ресурсів і підвищуючи ефективність моделей машинного навчання. Правильно побудована архітектура сприяє створенню гнучких систем, які можна налаштовувати й розширювати відповідно до потреб користувачів. У статті [1] показано, що перевагами використання методів машинного навчання є прискорення та здешевлення оцінки вартості землі.

У роботі [7] описані такі етапи (потреби) для успішного аналізу даних за допомогою алгоритмів машинного навчання: збір даних (collect), переміщення та зберігання (move/store), дослідження та трансформація (explore/transform), агрегування та розбивка (aggregate/label), вивчення й оптимізація даних (learn/optimize), машинне навчання.

**Збір, переміщення та зберігання даних**, **агрегування та розбивка**

Збір: як описано в [1], існує декілька можливих джерел даних – державні реєстри, інформація, яка доступна на сайтах оголошень чи аукціонів тощо. На цьому етапі визначимо джерела, інструментарій для збору даних, необхідність отримання звітів і журналів щодо автоматичного збору даних. Виконаємо автоматично збір даних окремо з кожного джерела, якщо вони мають відмінності у форматі збереження даних, частоти оновлення, об'єму тощо.

Переміщення та зберігання: зібрані дані збережемо в локальному середовищі. Для даних у файловому форматі csv (файл із даними, розділеними комою чи символом) є два можливі сценарії зберігання. Перший – завантажити в оперативну пам'ять, де виконуватиметься машинне навчання; другий – перенести інформацію з файлу в локальне сховище. Перевагою переносу даних в певну базу є наявність потужних можливостей щодо роботи з даними в сучасних СКБД. Для зберігання даних зручно використовувати бази даних, що підтримують SQL, або бази NoSQL. У випадку NoSQL код для інтеграції даних здається дещо простішим (через відсутність обов'язкової типізації, створення таблиць), але потребує контролю за джерелом даних, тому що набір даних (атрибутів, що описують земельну ділянку) змінюється, а отже, модифікується код, що збирає дані.

Дослідження та трансформація: проаналізуємо зібрані дані, їх структуру, потенційні взаємозв'язки, перевіримо на достовірність, видалимо некоректні дані (приміром, ціна продажу менша за нормативно-грошову оцінку землі, а на цей час такі продажі недозволені). Далі трансформуємо дані у зручний для подальшого використання формат: приведемо до єдиного формату вартість, дати покупки-продажу ділянки. Розширимо отриманий набір даними, які суттєво впливають на вартість землі (типом ґрунту, від якого залежить родючість землі, відстанню від великих міст), врахуємо інфляцію на дату оцінки. Встановлення типу ґрунту автоматизується простим алгоритмом, який визначає колір за певними координатами на карті й перетворює його у відповідний атрибут «тип ґрунту». Зручно користуватися кольоровою картою «Ґрунти України» [9] для визначення типу ґрунтів через її високу контрастність. Обрахуємо в кілометрах відстані від великих міст (адміністративних центрів областей) до ділянки за допомогою програмного коду, який вираховує відстань за формулою [10]:

$$\mathbf{distance} = \mathbf{arccos}(\mathbf{sin}(\mathbf{latitude}_1) * \mathbf{sin}(latitude_2) + \mathbf{cos}(latitude_1) * \mathbf{cos}((latitude_2) * cos(\mathbf{longitude}_1 - longitude_2)) * \mathbf{6371} \quad (3)$$

де 6371 – це радіус Землі в км, distance – значення відстані, latitude та longitude – відповідні значення координат.

Обрахунок нової ціни ділянки з урахуванням інфляції від моменту продажу розрахуємо за формулою:

$$Price_{new} = Price_{old} * \left(1 + \sum_{k=2022}^{2023} days_k * avgInfPerDay_k\right) \quad (4)$$

де $Price_{new}$ – нова ціна для машинного навчання, $Price_{old}$ – ціна на дату реєстрації права на земельну ділянку, $days_k$ – кількість днів, які вже пройшли в k-му році, $avgInfPerDay$ – середня інфляція за один день у k-му році. Значення інфляції візьмемо з [11].

Виконаємо категорійне кодування: переведемо текстові значення атрибутів у кількісні в числовому форматі. Із двох широко відомих технік категорійного кодування: кодування міток (label encoding) та кодування в унітарному коді (one hot encoding) – виберемо останнє, оскільки складніше зіставляти атрибути, задані у строковій формі.

Агрегування та розбивка: вибрані дані збираємо в єдиний набір і розбиваємо на тренувальний і тестовий.

Вивчення й оптимізація даних: досліджуємо використання різних методів машинного навчання, встановлюємо впливові атрибути для моделі.

За наявності N джерел даних на рис. 1 зображено відносини між застосунками, потрібними для реалізації вказаних вище етапів. Джерела даних $\{S_1, S_2, S_3, \ldots, S_N\}$ вимагають в загальному випадку N застосунків. Під час збирання даних ці застосунки виконують перевірку коректності ціни, площі ділянки в припустимих межах тощо. Також на рисунку зображено різні локальні сховища, де зберігаються дані з різних джерел (кількість сховищ та їх зв'язок із застосунками для збору даних залежить від особливостей програмної реалізації). Застосунок для агрегування використовує дані з локальних сховищ і створює єдиний набір із додатковими атрибутами (тип ґрунту, відстань від великих міст та інфляція за час до моменту оцінки). Застосунок для навчання використовує цей набір для отримання моделі обрахунку вартості земельної ділянки у вигляді, доступному для подальшого використання.

Якщо через користувацький інтерфейс програмного забезпечення ввести атрибути будь-якої ділянки, то програма покаже її вартість, яка є оціночною вартістю.

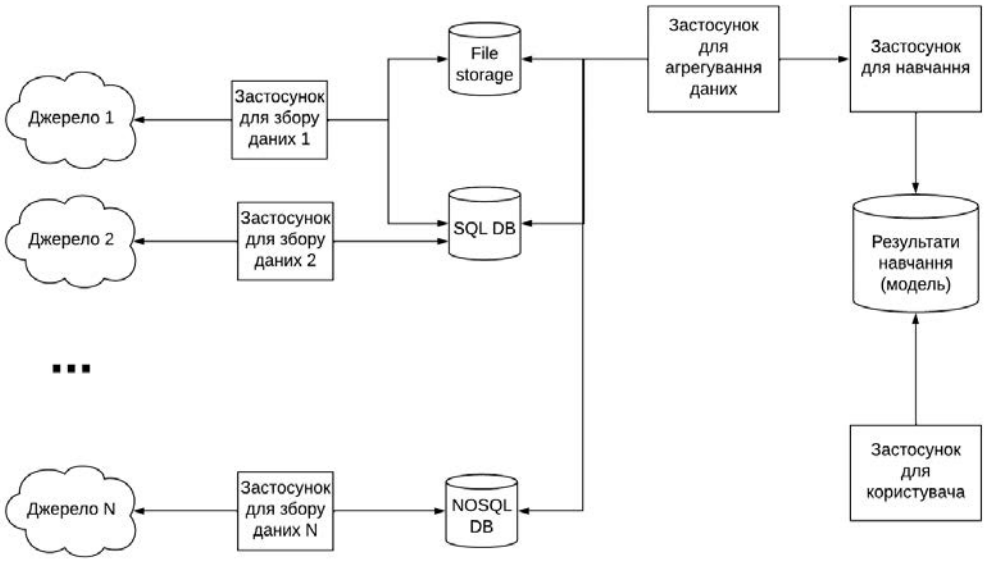


Рис. 1. Загальна архітектура програмного забезпечення для збору й аналізу даних про вартість землі сільськогосподарського призначення з використанням машинного навчання

Наведена архітектура дає змогу використовувати повністю автоматизовані застосунки для збору даних. У разі потреби можна додавати дані вручну за допомогою користувацького інтерфейсу.

Схожим чином, як у статті [1], зібрані дані з аукціонів, сайтів оголошень, тощо (перше джерело даних). Для 2 123 ділянок зібрано інформацію, що містить їх вартість, вартість оренди, термін оренди, НГО (нормативно-грошову оцінку), кадастровий номер, адміністративну приналежність, географічні координати, площу, прогнозовану родючість, тип ґрунту. Інші дані, які описують інформацію щодо продажу земельних ділянок, також є на сайті [2] (друге джерело даних). Станом на 01.01.2024 файл містить дані про 815 529 транзакцій, а саме: кадастровий номер земельної ділянки, КОАТУУ (класифікатор об'єктів адміністративно-територіального устрою України), площу ділянки, область, район, населений пункт, вулицю, назву територіальної громади, цільове призначення, назву угіддя, тип угоди. Цей перелік полів менший, ніж доступно в першому джерелі даних: відсутня інформація

про оренду, тип ґрунту та родючість. У файлі частина даних належить до транзакцій щодо спадщини й інших випадків, коли частина транзакцій не є купівлею-продажом ділянки. Для 197 886 ділянок із типом угоди купівлі-продажу лише у 137 358 випадках вказано ціну продажу. Ціна деяких ділянок або зависока, або занадто низька. Тобто, можливо, існують певні особливості ділянки (наявність корисних копалин із дозволом держави на їх видобуток тощо). За 2022 рік у файлі доступна інформація про 21 287 угод купівлі-продажу, а за 2023 рік – про 81 646 таких угод.

Створимо такі набори даних із двох джерел:

– з першого джерела виділимо повний набір усіх даних D1; D2 – множина D1 без даних про оренду, D3 – множина D1 з урахуванням інфляції в ціні земельної ділянки. Загальна кількість рядків у кожному наборі даних D1–D3 – 2 123 (1 486 у тренувальному та 637 у тестовому);

– із другого джерела виділимо такі набори даних: D4 – підмножина повного набору із застосуванням фільтрів за датою за 2022 та 2023 роки, типу «договір купівлі-продажу», і тільки ділянки, для яких вказана ціна продажу. Загальна кількість рядків у наборі даних D4 – 102 924; D5 – підмножина D4 для ділянок, вартість за 1 га яких не менше ніж 10 000 грн та не більше ніж 150 000 грн. Кількість ділянок у наборі D5 становить 82 241. D6 – множина D5, розширена інформацією про відстані від найбільших міст України до земельної ділянки. D7 – множина D6, розширена інформацією про інфляцію.

Кожен набір даних розбитий на два: тренувальний і тестовий у співвідношенні 7 : 3. Дані відсортовані випадковим чином, але в різних наборах даних збережене розподілення між тренувальними та тестовими даними. Тобто, якщо у наборі D1 певна ділянка потрапила до тренувального набору даних, то і в D2 ця ділянка буде в тренувальному наборі.

**Виконання досліджень**

З використанням 7 наборів даних, які виділені з двох джерел, проведемо машинне навчання. Відмінністю в порівнянні з [1] буде врахування даних про інфляцію та використання даних від Державної служби України з питань геодезії, картографії та кадастру [2].

Як мову програмування виберемо мову C# та фреймворк машинного навчання ML.NET. У цьому фреймворку доступні методи машинного навчання LightGBM (цей метод, як і XGBoost, належить до методів градієнтного бустінгу – підсилення), Fast Tree, Fast Forest.

Проведемо навчання, оптимізуючи моделі за коефіцієнтом детермінації ( $R^2$ ).

Результати навчання наведено в таблиці 1.

Таблиця 1

Результат машинного навчання, коефіцієнт детермінації

| Номер набору даних | $R^2$ | | |
|---|---|---|---|
| | LightGBM | Fast Tree | Fast Forest |
| D1 | 0,851 | 0,897 | 0,648 |
| D2 | 0,543 | 0,58 | 0,56 |
| D3 | 0,874 | 0.862 | 0,660 |
| D4 | 0,003 | 0,006 | 0,004 |
| D5 | 0,259 | 0,345 | 0,294 |
| D6 | 0,314 | 0,328 | 0,295 |
| D7 | 0,337 | 0,368 | 0,301 |

Для визначення впливовості певного атрибуту в наборі даних розрахуємо значення важливості індексу перестановки PFI (Permutation Feature Importance). У таблиці 2 наведено впливовість атрибутів для набору даних D1. Значення впливовості дорівнює індексу перестановки.

Таблиця 2

Впливовість атрибутів для набору даних D1

| Індекс | Впливовість атрибута | | |
|---|---|---|---|
| | LightGBM | Fast Tree | Fast Forest |
| 1 | Area (0,84) | area (1,06) | dystanceFromOdessa (0,28) |
| 2 | rentalYield (0,8) | rentalYield (0,96) | rentRate (0,13) |
| 3 | rentRate (0,77) | rentRate (0,92) | rentalYield (0,11) |
| 4 | dystanceFromOdessa (0,26) | dystanceFromOdessa (0,11) | coordinateLatitude (0,09) |
| 5 | dystanceFromKrivoiRog (0,09) | distanceFromKyiv (0,04) | distanceFromKyiv (0,04) |

На рис. 2 відображено відношення PFI для п'яти найвпливовіших атрибутів для LightGBM.

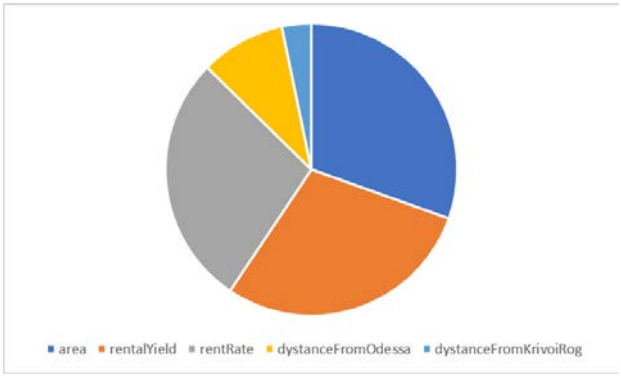


Рис. 2. Найвпливовіші атрибути для LightGBM

На рис. 3 відображено відношення PFI для п'яти найвпливовіших атрибутів для Fast Tree.

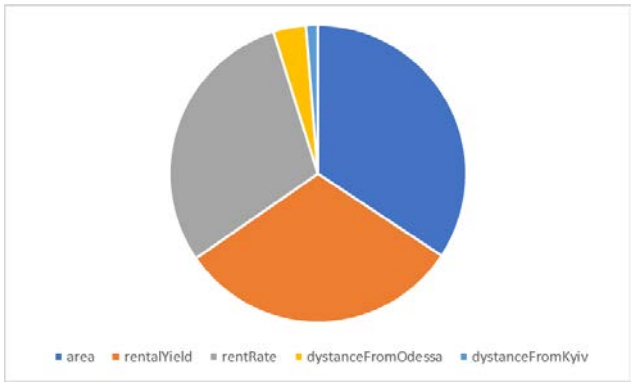


Рис. 3. Найвпливовіші атрибути для Fast Tree

На рис. 4 відображено відношення PFI для п'яти найвпливовіших атрибутів для Fast Forest.

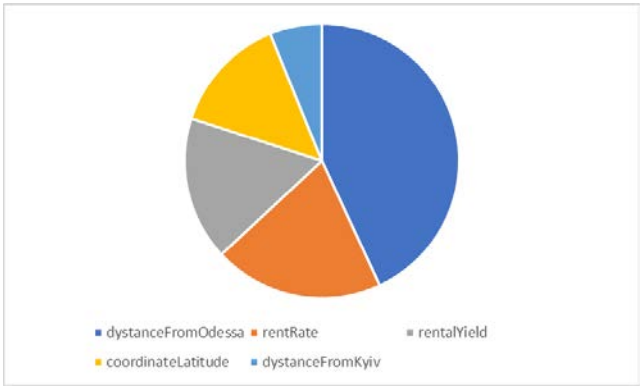


Рис. 4. Найвпливовіші атрибути для Fast Forest

У таблицях 3–5 наведено відсоток коректних передбачень згідно з формулою (2) для тестового набору даних.

Таблиця 3
Відсоток коректних передбачень – LightGBM

| **Номер набору даних** | **D1** | **D2** | **D3** | **D4** | **D5** | **D6** | **D7** |
|---|---|---|---|---|---|---|---|
| Відсоток коректний передбачень | 84 | 55 | 88 | 2 | 21 | 18 | 59 |

Таблиця 4
Відсоток коректних передбачень – Fast Tree

| Номер набору даних | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|
| Відсоток коректний передбачень | 89 | 38 | 91 | 23 | 29 | 15 | 59 |

Таблиця 5
Відсоток коректних передбачень – Fast Forest

| Номер набору даних | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|
| Відсоток коректний передбачень | 43 | 36 | 47 | 9 | 13 | 8 | 56 |

Наведені результати показують, що на якість моделі значно впливає наявність інформації про оренду (D2 має гірші результати, ніж D1). Врахування інфляції за два роки покращує значення <<Eqn022.eps>> на 2–4% для першого набору даних і на 2–11% – для другого. З додаванням даних про відстані від великих міст і з урахуванням рівня інфляції для другого набору даних якість моделі покращується в декілька разів (D7 має набагато кращий відсоток коректних передбачень, ніж D4). При цьому додавання лише інформації про відстані чи лише врахування інфляції не дає значного приросту коректного передбачення (порівняння D4 та D5, D4 та D6). У певних випадках (D1 і D3, метод Fast Tree; D5 і D6, метод LightGBM) коефіцієнт детермінації погіршується, а відсоток коректних передбачень незначно покращується. Якість моделі для другого джерела даних нижча, ніж для першого. Найгірші результати отримано для D4 через наявність даних, що здаються недостовірними і які були видалені для наборів D5–D7.

У наборах D3 та D7 621 ділянка має однаковий кадастровий номер. З них для моделі D3 і моделі D7 передбачено коректно 573 та 421 випадків відповідно, тобто відсоток коректних передбачень становив 92% та 67%. Цей відсоток кращий, ніж наведено в таблицях 3–5, бо частина із 621 ділянки є в навчальному наборі даних.

**Висновки**

У роботі наведено загальну схему архітектури програмного забезпечення для збору й аналізу даних про вартість землі сільськогосподарського призначення з використанням машинного навчання, яка охоплює чотири різні типи застосунків: для збору даних, для агрегування даних, для навчання та застосунок для користувача. Отримано результати роботи моделі передбачення для даних із двох джерел. Відсоток коректних передбачень залежно від набору даних і вибраного алгоритму сягає 91% для тестових наборів даних. Найкраще значення коефіцієнта детермінації залежно від набору даних і вибраного алгоритму становить 0,897, що є досить гарним результатом для подальшої роботи в цьому напрямі. Набір даних, що містить інформацію про ціни з урахуванням інфляції за два останні роки, на 4% краще передбачає вартість землі. Також, на відміну від [1], на перше місце за значенням <<Eqn023.eps>> вийшов алгоритм Fast Tree. Додавання інформації про відстані від великих міст до набору даних із другого джерела покращило коефіцієнт детермінації на 20% (з 0,259 до 0,314). Якість передбачення для другого джерела гірша через меншу кількість атрибутів і відсутність інформації про оренду, термін оренди, родючість ґрунтів, деякі неточності в даних (приміром, неринкова ціна за договором купівлі-продажу, тобто менша за НГО). Кращим результатом для другого джерела є коефіцієнт детермінації 0,368 із відсотком коректних передбачень 56%. Застосування другого джерела показує, що за допомогою додавання нових атрибутів (тип ґрунту, відстань до ділянки, інфляція) точність передбачення значно покращується.

У порівнянні з [1] отримано модель, яка краще відповідає поставленій задачі знаходження достовірної оцінки вартості земельної ділянки. Відсоток коректних передбачень для тестового набору даних збільшився із 77 до 91%, коефіцієнт детермінації зріс з 0,818 до 0,897.

Через те що на ціну можуть вплинути певні неринкові чинники (забрудненість ділянки вибухонебезпечними предметами і т. ін.), подальша інтеграція додаткових джерел даних (приміром, карти ДСНС України «Інтерактивна мапа територій, які потенційно можуть бути забруднені вибухонебезпечними предметами» [12]) підвищить точність оцінки.

***Література***

1. Бутенко М. Є., Павленко В. І. Передбачення вартості землі сільськогосподарського призначення за допомогою алгоритмів машинного навчання. Інфокомунікаційні та комп'ютерні технології. 2023. № 1. С. 161–167.

2. Інформація про здійснені транзакції по земельним ділянкам за період, розділ «Моніторинг земельних відносин». https://land.gov.ua/. URL: https://land.gov.ua/ (дата звернення: 10.01.2024).

3. Machine-Learning-Based Prediction of Land Prices in Seoul, South Korea / K. Jungsun et al. Sustainability. 2021. No. 13 (23).

4. Naotunna R. A Model for the Estimation of Land Prices in Colombo District using Web Scraped Data. University of Colombo School of Computing. 2022.

5. Yunjong K., Seungwoo C., Mun Yong Y. Applying Comparable Sales Method to the Automated Estimation of Real Estate Prices. Sustainability. 2020. No. 12 (14).

6. Extreme gradient boosting-based machine learning approach for green building cost prediction / O. Alshboul et al. Sustainability. 2022. No. 14.

7. Rogati M. The AI Hierarchy of Needs. https://medium.com/hackernoon/the-ai-hierarchy-of-needs-18f111fcc007. URL: https://medium.com/hackernoon/the-ai-hierarchy-of-needs-18f111fcc007 (дата звернення: 11.03.2024).

8. Ranjit K. Machine learning techniques for forecasting agricultural prices: A case of brinjal in Odisha, India. Plos one. 2022. No. 17(7).

9. Ґрунти України. Карта. За ред. М. І. Полупан, В. Б. Соловей, В. А. Величко. Національний науковий центр «Інститут ґрунтознавства та агрохімії ім. О. Н. Соколовського», 2005.

10. Houde K.V., Hegadi R.S. Optimizing Drone Navigation Using Shortest Path Algorithms. International Conference on Recent Trends in Image Processing and Pattern Recognition. 2023.

11. Коментар Національного банку щодо рівня інфляції у 2022 році. https://bank.gov.ua/ua/news/all/komentar-natsionalnogo-banku-schodo-rivnya-inflyatsiyi-u-2022-rotsi. URL: https://bank.gov.ua/ua/news/all/komentar-natsionalnogo-banku-schodo-rivnya-inflyatsiyi-u-2022-rotsi (дата звернення: 11.03.2024).

12. Інтерактивна мапа територій, які потенційно можуть бути забруднені вибухонебезпечними предметами. https://mine.dsns.gov.ua/. URL: https://mine.dsns.gov.ua/ (дата звернення: 11.03.2024).

### *References*

1 Butenko, M., Pavlenko, V. (2023). Agricultural land price prediction using machine learning algorithms. Infocommunication and computer technologies. № 1. P. 161–167.

2. Information on transactions carried out on land for the period is available in the "Land Relations Monitoring" section on the website https://land.gov.ua/. [URL: https://land.gov.ua/] (date of access: 10.01.2024).

3. Machine-Learning-Based Prediction of Land Prices in Seoul, South Korea (2021). K. Jungsun et al. Sustainability. No. 13 (23).

4. Naotunna R. (2022). A Model for the Estimation of Land Prices in Colombo District using Web Scraped Data. University of Colombo School of Computing.

5. Yunjong, K., Seungwoo, C., Mun Yong, Y. (2020). Applying Comparable Sales Method to the Automated Estimation of Real Estate Prices. Sustainability. No. 12 (14).

6. Extreme gradient boosting-based machine learning approach for green building cost prediction (2022). O. Alshboul et al. Sustainability. No. 14.

7. Rogati, M. The AI Hierarchy of Needs. https://medium.com/hackernoon/the-ai-hierarchy-of-needs-18f111fcc007. URL: https://medium.com/hackernoon/the-ai-hierarchy-of-needs-18f111fcc007 (date of access: 11.03.2024).

8. Ranjit, K. (2022). Machine learning techniques for forecasting agricultural prices: A case of brinjal in Odisha, India. Plos one. No. 17 (7).

9. Soils of Ukraine. Map. (2005). Ed. M.I. Polupan, V.B. Solovey, V.A. Velychko. National Scientific Center "Institute of Soil Science and Agrochemistry named after O.N. Sokolovsky".

10. Houde, K.V., Hegadi, R.S. (2023). Optimizing Drone Navigation Using Shortest Path Algorithms. International Conference on Recent Trends in Image Processing and Pattern Recognition.

11. Commentary of the National Bank on the inflation level in 2022. URL: https://bank.gov.ua/en/news/all/komentar-natsionalnogo-banku-schodo-rivnya-inflyatsiyi-u-2022-rotsi (accessed: 11.03.2024).

12. Interactive map of areas potentially contaminated with explosive objects. https://mine.dsns.gov.ua/. URL: https://mine.dsns.gov.ua/ (date of access: 11.03.2024).

**УДК 004.623**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-04**

# ІНФОРМАЦІЙНА СИСТЕМА ЗБОРУ ТА НАКОПИЧЕННЯ ДАНИХ ПРО ЯКІСТЬ АТМОСФЕРНОГО ПОВІТРЯ ЗІ СТАНЦІЙ VAISALA МУНІЦИПАЛЬНОГО РІВНЯ

***Вадурін К. О.**, аспірант, Кременчуцький національний університет імені Михайла Остроградського, Кременчук, Україна*
*https://orcid.org/0000-0001-7781-5783, kir3337@gmail.com*

***Перекрест А. Л.**, д-р техн. наук, професор, Кременчуцький національний університет імені Михайла Остроградського, Кременчук, Україна*
*https://orcid.org/0000-0002-7728-9020, pksg13@gmail.com*

***Бахарєв В. С.**, д-р техн. наук, професор, Кременчуцький національний університет імені Михайла Остроградського, Кременчук, Україна*
*https://orcid.org/0000-0001-9312-654X, v.s.baharev@gmail.com*

***Дерієнко А. І.**, канд. техн. наук, доцент, директор ТОВ «ЛЕМПДЕВ», Кременчук, Україна*
*andrey.deriyenko@gmail.com*

***Іващенко А. В.**, студент, Кременчуцький національний університет імені Михайла Остроградського, Кременчук, Україна*
*a.ivascheko2020@gmail.com*

***Шкарупа С. А.**, магістр, співробітник ТОВ «ЛЕМПДЕВ», Кременчук, Україна*
*sergii.shkarupa@gmail.com*

***Анотація.** На сьогодні проблема забруднення атмосферного повітря актуальна, особливо в Україні. Унаслідок війни та катастрофи на Каховській ГЕС, а також спричинених ними перенесення окремих промислових підприємств і переміщення населення виникли значні зміни у стані атмосферного повітря муніципалітетів порівняно зі станом до цих змін, що фіксувався постами екологічного моніторингу та міг прогнозуватися на довгий час. У Кременчуці дослідження стану атмосферного повітря здійснюються комунальним підприємством міської ради. До автоматичного обладнання підприємства входять декілька станцій Vaisala, які підтримують вивід даних за допомогою API у сервіс, що розповсюджується за підписним принципом. Базовий сервіс може зберігати дані від станцій, будувати графіки й виводити показники у CSV-форматі, але не може бути вдосконалений для формування звітності за внутрішніми стандартами документообігу та не підтримує формування звітів за постановою № 827 КМУ.*

*Тому метою цього дослідження є розробка та впровадження інформаційної системи з автоматичного збору, накопичення, структурування даних зі станцій Vaisala й автоматизованого формування звітних відомостей про стан атмосферного повітря, що зменшить час, потрібний оператору для підготовки та надання вичерпної інформації про можливі перевищення шкідливих домішків у повітрі регулюючим органам, виходячи зі стандартів внутрішнього документообігу. Спочатку в дослідженні проаналізовано подібні вебресурси, що надають дані про стан атмосферного повітря та є у відкритому доступі. Потім розроблено й реалізовано інформаційну систему, що базується на ОС Ubuntu, сервері Nginx, системі керування базами даних MySQL, мові PHP з фреймворками PHPWord та Laravel, бібліотеками JavaScript Vue.js, Bootstrap та мережевих технологіях vaisala_api, SPA, AJAX. Сервер модифіковано для обробки великого об'єму даних зі станцій Vaisala, для яких створені окремі таблиці для вивантаження «сирих» даних і таблиця для перерахунку хвилинних показів у 20-хвилинні, денні, місячні та річні.*

***Ключові слова:** якість атмосферного повітря, екологічний моніторинг, Nginx, Vaisala, Laravel, VPS.*

# AN INFORMATION SYSTEM FOR COLLECTING AND STORING AIR QUALITY DATA FROM MUNICIPAL LEVEL VAISALA STATIONS

***Kyrylo Vadurin**, Postgraduate Student, Kremenchuk Mykhailo Ostrohradskyi National University, Kremenchuk, Ukraine*
*https://orcid.org/0000-0001-7781-5783, kir3337@gmail.com*

***Andrii Perekrest**, Dc.S., Prof., Kremenchuk Mykhailo Ostrohradskyi National University, Kremenchuk, Ukraine*
*https://orcid.org/0000-0002-7728-9020, pksg13@gmail.com*

***Volodymyr Bakharev**, Dc.S., Prof., Kremenchuk Mykhailo Ostrohradskyi National University, Kremenchuk, Ukraine*
*https://orcid.org/0000-0001-9312-654X, v.s.baharev@gmail.com*

***Andriy Deriyenko**, PhD, Ass. Prof., Director of LEMPDEV LLC, Kremenchuk, Ukraine*
*andrey.deriyenko@gmail.com*

***Artem Ivashchenko**, Student, Kremenchuk Mykhailo Ostrohradskyi National University, Kremenchuk, Ukraine*
*a.ivascheko2020@gmail.com*

***Sergii Shkarupa**, Employee of LEMPDEV LLC, Kremenchuk, Ukraine*
*sergii.shkarupa@gmail.com*

***Abstract.** Today, the problem of air pollution is a pressing one, especially in Ukraine. As a result of the war and the Kakhovka hydroelectric power station disaster, as well as the relocation of certain industrial enterprises and the displacement of the population, there have been significant changes in the state of the atmospheric air in municipalities compared to the state before these changes, which was recorded by environmental monitoring stations and could be predicted for a long time. In Kremenchuk, air quality studies are carried out by a municipal enterprise of the city council. The company's automatic equipment includes several Vaisala stations that support data output via API to a subscription-based service. The basic service can store data from the stations, build graphs and output indicators in CSV format, but cannot be upgraded to generate reports according to internal document management standards and does not support the generation of reports in accordance with CMU Resolution No. 827.*

*Therefore, the purpose of this study is to develop and implement an information system for the automatic collection, accumulation, structuring of data from Vaisala stations and automated generation of reports on the state of the atmospheric air, which will reduce the time required for the operator to prepare and provide comprehensive information on possible exceedances of harmful impurities in the air to regulatory authorities based on internal document management standards. First, the study analysed similar web resources that provide data on the state of the air and are publicly available. Then we developed and implemented an information system based on Ubuntu OS, Nginx server, MySQL database management system, PHP language with PHPWord and Laravel frameworks, Vue.js and Bootstrap JavaScript libraries, and vaisala_api, SPA, and AJAX network technologies. The server has been modified to process a large amount of data from Vaisala stations, for which separate tables have been created to upload raw data and a table to convert minute readings into 20-minute, daily, monthly and annual readings.*

***Key words:** air quality, environmental monitoring, Nginx, Vaisala, Laravel, VPS.*

**Вступ**

Сьогодні проблема забруднення атмосферного повітря актуальна як для світу, так і для Україні. Війна, знищення Каховської ГЕС, руйнування ТЕЦ, нафтопереробних заводів, ліній живлення атомних станцій, а також спричинені війною релокації бізнесу та населення викликають значні зміни екології муніципалітетів, що потребує наскрізного моніторингу всіх маркерів якості атмосферного повітря. У Кременчуці дослідженнями стану атмосферного повітря займається комунальне підприємство «Науковий центр еколого-соціальних досліджень» Кременчуцької міської ради Кременчуцького району Полтавської області, що має на балансі автоматичні й автоматизовані засоби для моніторингу. До автоматичного обладнання підприємства належать станції Vaisala, які підтримують вивід даних за допомогою API. Також станції можуть виводити дані в сервіс виробника, який розповсюджується за підписним принципом і надає доступ до завантаження даних і формування за ними графіків. Але базовий сервіс не піддається модифікації через закритість коду серверної частини, і вартість підписки є достатньо високою для фінансування з бюджетних коштів.

Тому метою цього дослідження є розробка та впровадження інформаційної системи з автоматичного збору, накопичення, структурування даних зі станцій Vaisala й автоматизованого формування звітних відомостей про стан атмосферного повітря, що зменшить час, потрібний оператору для

підготовки й надання вичерпної інформації про можливі перевищення шкідливих домішок у повітрі регулюючим органам, виходячи зі стандартів внутрішнього документообігу.

**Виконання досліджень**

За результатами аналізу матеріалів, доступних у системах пошуку за наукометричними базами [1–3] та доступних через стандартні пошукові системи [4–5] у відкритому доступі не виявлено готових наукових чи технічних вебрішень для автоматизованого формування текстових звітів про дані досліджень з постів екологічного моніторингу.

Під час аналізу матеріалів в Інтернеті, наявних у відкритому доступі, виявлено велику кількість інформаційних вебсервісів із графічною індикацією показів з екологічних постів для інформування населення, а також локальні сторінки екологічного моніторингу уряду найбільших розумних міст Землі, на які і будемо орієнтуватися під час створення науково-технічного рішення.

Проєкт World Air Quality Index (WAQI) [6] надає прозору інформацію про якість повітря для громадян світу.

Розробники проєкту WAQI стверджують, що завжди прагнуть надавати історичні дані про якість повітря установам і організаціям, які працюють у сфері екологічної обізнаності, моніторингу якості повітря, а також медичних та епідеміологічних досліджень.

Метою бази відкритих даних є надання необмеженого та безкоштовного доступу до світових історичних даних і даних про якість повітря в реальному часі з понад 100 країн, охоплених на aqicn.org.

На ресурсі також доступні середньодобові дані про якість повітря для всіх станцій, які можна безкоштовно завантажити з форми. Слід зазначити, що всі цифри конвертуються в AQI за стандартом US EPA.

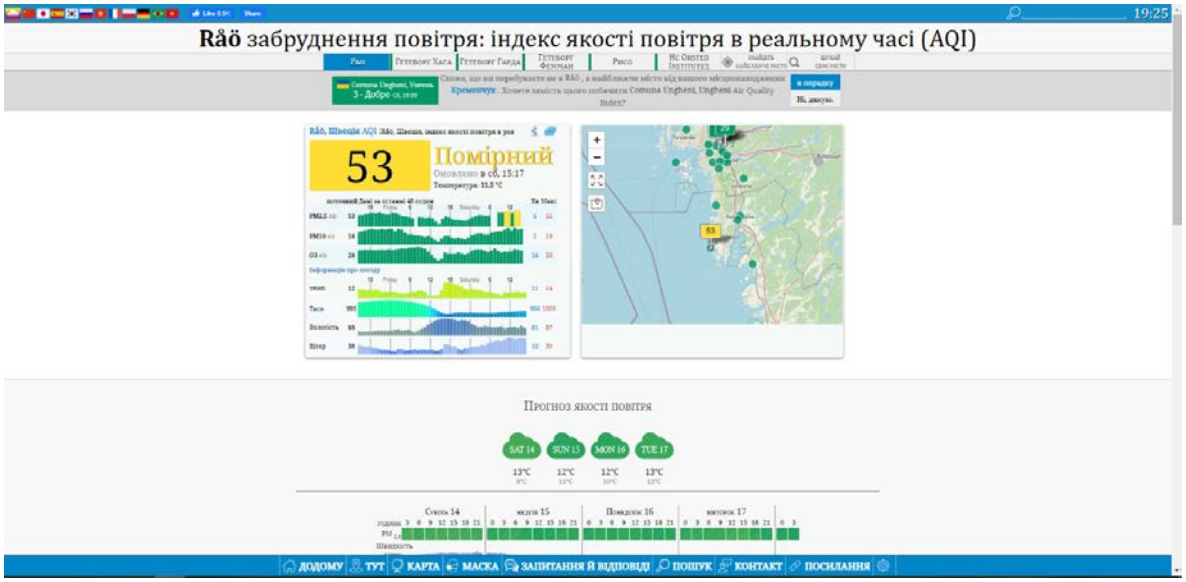


Рис. 1. Інтерфейс ресурсу WAQI

BreezoMeter [7] використовує дані CAMS у своїх алгоритмах. Багаторівневий підхід починається з державних станцій моніторингу, які надають дуже точну інформацію з низькою роздільною здатністю, оскільки станції зазвичай не розташовані близько одна до одної. Комбінуючи різні джерела інформації, BreezoMeter може перевірити їх, порівнюючи один з одним. CAMS надає безперервну інформацію про такі забруднювачі, як тверді частинки, діоксид сірки, діоксид азоту, чадний газ та озон, тоді як деякі місцеві станції зазвичай вимірюють лише один або два із цих забруднювачів. Крім того, є величезні ділянки землі, де немає наземних станцій, а дані CAMS (у поєднанні з іншими даними) дають можливість розширити охоплення, надаючи дані про якість повітря в малонаселені райони. Таким чином, включення даних CAMS дає змогу BreezoMeter пропонувати теплові карти забруднення в реальному часі з вищою роздільною здатністю. Цей внесок в одну з найпопулярніших функцій продукту важливий для користувачів, які намагаються отримати візуальне уявлення про забруднення повітря, те, що інакше зазвичай важко або неможливо побачити.

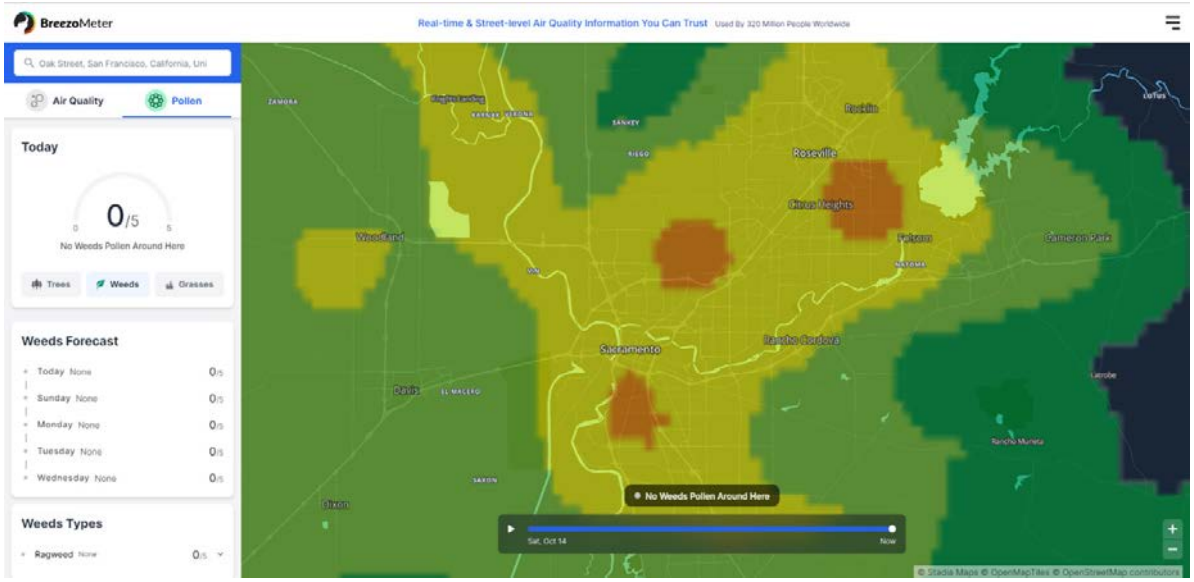


Рис. 2. Інтерфейс ресурсу BreezoMeter

AirNow [8] – це єдине джерело даних про якість повітря. Сайт висвітлює якість повітря в місцевості, водночас надаючи інформацію про якість повітря в державному, національному та світовому масштабах. Нова інтерактивна карта дає змогу зменшити масштаб, щоб отримати загальну картину, або деталізувати її, щоб переглянути дані для одного монітора якості повітря.

AirNow звітує про якість повітря за допомогою офіційного індексу якості повітря США (AQI), кольорового індексу, призначеного для визначення того, чи якість повітря є задовільною чи ні.

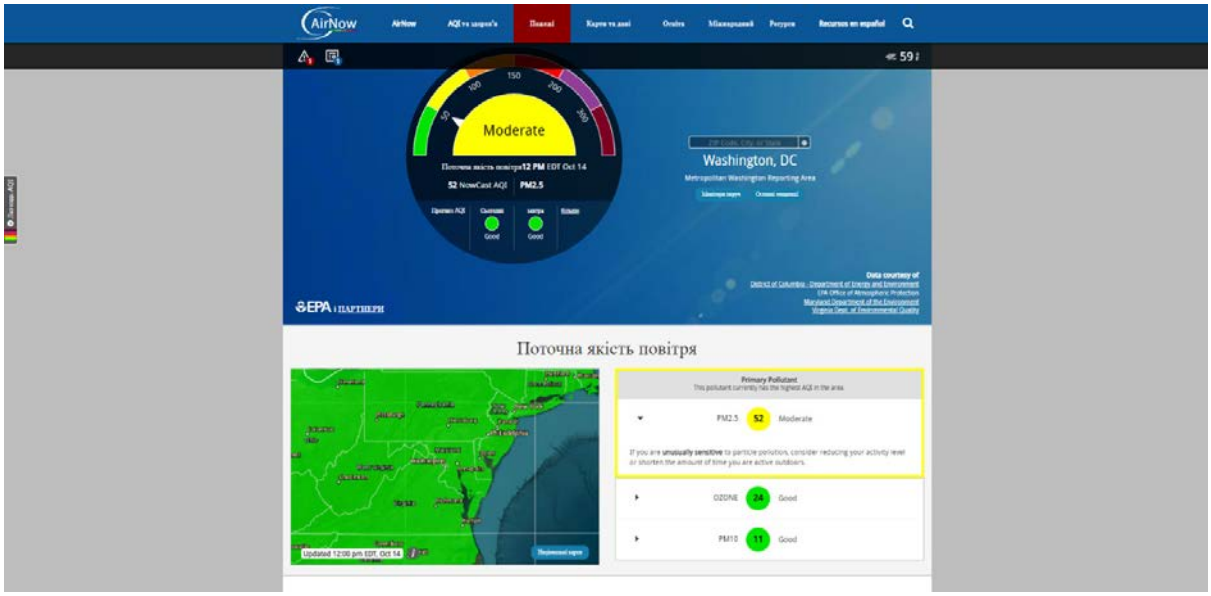


Рис. 3. Інтерфейс ресурсу AirNow

Централізована система даних AirNow забезпечує контроль якості, послідовність національних звітів і можливість поширювати дані серед громадськості, дослідників, компаній та інших систем даних.

Глобальний метеорологічний центр AccuWeather 6 травня 2020 р. оголосив про те, що інтегрував життєво важливі дані про якість повітря й аналітику у свої прогнози погоди на сайті accuweather.com [9]. Дані отримані завдяки ексклюзивному партнерству з паризькою лабораторією Plume Labs, головною метою якої є надання інформації про якість повітря доступною та розширення можливостей людей у всьому світі для кращого контролю за своїм здоров'ям, і вони недоступні з будь-якого іншого джерела.

За результатами аналізу представлених вебресурсів, що є у відкритому доступі, виявлено, що наразі відсутнє відкрите рішення для автоматизованого формування звітів про якість повітря в текстовому форматі Word.

У деяких ресурсів наявна функція для завантаження набору знятих усереднених даних у стандартизованих одинцях виміру чи перерахованих на власні індекси якості повітря. На всіх ресурсах доступна мапа з відміченими координатами розміщення станцій моніторингу, з натисканням на які можна отримати доступ до показників станції, що виводяться згідно з регламентом надавання послуг відповідного ресурсу.

Для розгортання сервера вибрано VPS хостинг Zomro.com [10].

Zomro.com – голландська (Zomro BV) компанія, що надає безліч послуг, включно зі спільним хостингом, VDS/VPS і виділеними серверами, реєстрацією доменів і додатковими параметрами хостингу.

Для розгортання сервера з базою даних вибрано тарифний план VPS – Advanced, за якого пропонується 2 процесорні ядра Intel Xeon, 2 Гб оперативної пам'яті, 40 Гб дискового простору на SSD, ОС Linux з оплатою у 82,25 євро за 12 місяців користування (з урахуванням скидки).

Операційна система встановлена на сервері за замовчуванням Linux Ubuntu 20.04 LTS.

Оновлення до Ubuntu випускається кожні шість місяців, а довгострокова підтримка (LTS) – кожні два роки. Станом на жовтень 2023 року найновішим випуском є 23.10 Mantic Minotaur, а поточним довгостроковим випуском підтримки є 22.04 Jammy Jellyfish.

Виходячи із цієї інформації, версію, встановлену за замовчуванням, на сервері було оновлено до останньої стабільної версії 22.04 Jammy Jellyfish.

Програмне забезпечення, що використовується на сервері для організації його функціонування, наведено на рис. 4.

```
root@vm4652831:/var/www/EcoKremCity# nginx -v; mysql -V; php -v
nginx version: nginx/1.18.0 (Ubuntu)
mysql  Ver 8.0.34-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
    */10 * * * * cd /var/www/EcoKremCity; php artisan api:download-vaisala
    */20 * * * * cd /var/www/EcoKremCity; php artisan splits:vaisala
```

Рис. 4. Програмне забезпечення, що використовується на сервері

Nginx (версія 1.8.0 для Ubuntu) [11] – це вебсервер та проксісервер, написаний на мові програмування C, завдяки якому в проєкті можна забезпечити роботу веббраузера клієнта із сервером. Він призначений для обробки запитів HTTP, HTTPS, SMTP та інших мережевих протоколів. Один з основних принципів роботи Nginx – асинхронність та подвійне управління, що дає змогу обробляти багато одночасних запитів і забезпечувати високу продуктивність.

MySQL (версія 8.0.34 для Ubuntu) [12] – це система управління базами даних (СУБД), яка використовується для збереження, організації та маніпуляції структурованими даними в базі екологічних даних. Вона базується на мові запитів SQL (Structured Query Language) і використовується для взаємодії з базами даних на різних рівнях.

PHP (PHP: Hypertext Preprocessor, версія 8.1.2 для Ubuntu) [13] – це серверна мова програмування, яка широко використовується для розробки вебсайтів і вебдодатків.

Laravel [14] – це відкритий фреймворк для створення вебдодатків мовою програмування PHP. Він забезпечує програмістам готовий інструментарій і структуру для розробки високоякісних вебдодатків, прискорюючи процес розробки та полегшуючи підтримку.

PHPWord [15] – це бібліотека для створення й редагування документів у форматі Microsoft Word (DOCX) за допомогою мови програмування PHP. Ця бібліотека надає можливість генерувати документи з різноманітними елементами, такими як текст, таблиці, графіка, гіперпосилання й інші.

DBeaver [16] – це потужний і універсальний клієнт баз даних, призначений для роботи з різними системами управління базами даних (СУБД). Він надає інструменти для підключення, адміністрування, запитів і візуалізації даних у різних типах баз даних, включно з MySQL, PostgreSQL, Oracle, Microsoft SQL Server та багатьма іншими.

Cron – це стандартний планувальник завдань у багатьох операційних системах, особливо у UNIX-подібних системах. Він дає користувачам змогу створювати та планувати автоматичні завдання, які виконуватимуться в певний час або інтервал часу.

Node.js [17] – це вільна, відкрита технологія для створення серверних додатків на платформі JavaScript. Вона базується на движку V8, розробленому Google, і надає можливість виконувати JavaScript на стороні сервера.

Vue.js [18] – це прогресивна JavaScript-бібліотека, спроєктована для побудови вебінтерфейсів. Вона базується на архітектурному шаблоні Model-View-ViewModel (MVVM) та надає різноманітні інструменти й можливості для створення динамічних та інтерактивних веб-сторінок.

Bootstrap [19] – це відкритий фреймворк для веброзробки, який розроблений і підтримується Twitter. Він надає набір інструментів, компонентів та стилів для розробки зручних і респонсивних вебсайтів і вебдодатків.

Спочатку було розроблено базу даних, що містить таблиці для зберігання даних зі станцій EcoCity (для підтримки подальшого розширення інформаційної системи дешевшими станціями нижчого рівня моніторингу), а також окремі таблиці даних станцій Vaisala та загальну таблицю бази даних з 20-хвилинним, денним, річним усередненням vaisala_splits (рис. 5).


Рис. 5. Загальна розроблена структура бази даних

Основні таблиці бази даних за станціями Vaisala та загальна таблиця для формування автоматизованих звітів зазначено на рис. 6.

Моделі таблиць баз даних станцій зроблені універсальними для різних наборів датчиків, якщо в одній станції відсутній набір датчиків, то відповідні колонки будуть порожніми або заповненими нулями. Унікальним ідентифікатором у такій моделі є стовпець measurement_time, за допомогою якого реалізується алгоритм видалення дублікатів даних із таблиці. Слід відзначити, що перед усередненням даних і доданням їх до таблиці vaisala_splits дані щодо кожної станції містяться у вихідній таблиці з проміжком у 1 хв, тому вихідні таблиці можуть вимагати багато місця для зберігання даних.


Рис. 6. Таблиці даних, пов’язані зі станціями Vaisala

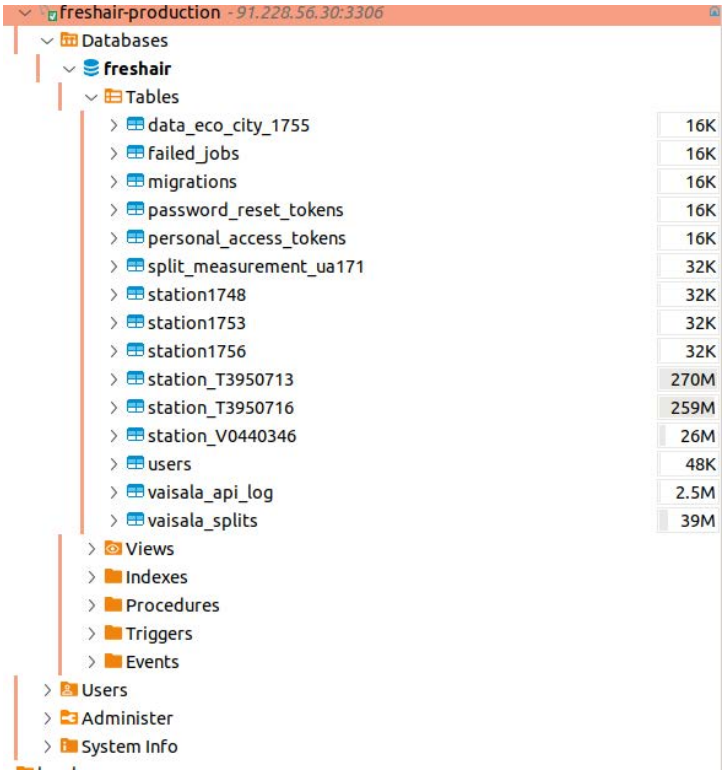


Рис. 7. Об’єм даних у таблицях

| id | humidity | temperature | pressure | carbon_oxide | nitrogen_dioxide | dust_PM2_5 | dust_PM10 | dust_PM1 | dust_PM_dioxide | hydrogen_sulfide | x_sulfur_wind | n_intensity | wind_direction | nd_speed | n_acc_um | created_at | updated_at | measurement_time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 872871 | 36.7 | 23.1 | 101610.0 | 0.0204 | 0.0003 | 9.9 | 88.7 | 4.7 | 0.0007 | -0.0002 | 4.2 | 0.0 | 113.9 | 1.4 | 0.0 | 11.10.2023 9:14 | 11.10.2023 9:15 | 25.09.2023 8:08 |
| 872870 | 37.1 | 22.9 | 101610.0 | 0.0205 | 0.0003 | 9.9 | 88.7 | 4.7 | 0.0007 | -0.0002 | 4.2 | 0.0 | 109.2 | 1.4 | 0.0 | 11.10.2023 9:14 | 11.10.2023 9:15 | 25.09.2023 8:07 |
| 872869 | 37.4 | 22.8 | 101620.0 | 0.0207 | 0.0003 | 9.9 | 88.7 | 4.7 | 0.0007 | -0.0001 | 4.2 | 0.0 | 107.3 | 1.4 | 0.0 | 11.10.2023 9:14 | 11.10.2023 9:15 | 25.09.2023 8:06 |
| 872868 | 38.1 | 22.7 | 101620.0 | 0.0205 | 0.0003 | 9.9 | 88.7 | 4.7 | 0.0007 | -0.0001 | 4.2 | 0.0 | 107.4 | 1.5 | 0.0 | 11.10.2023 9:14 | 11.10.2023 9:15 | 25.09.2023 8:05 |
| 872867 | 37.5 | 22.6 | 101620.0 | 0.0208 | 0.0003 | 9.9 | 88.7 | 4.7 | 0.0007 | -0.0001 | 4.2 | 0.0 | 107.6 | 1.5 | 0.0 | 11.10.2023 9:14 | 11.10.2023 9:15 | 25.09.2023 8:04 |
| 872866 | 38.2 | 22.4 | 101620.0 | 0.0207 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 4.2 | 0.0 | 107.4 | 1.3 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 8:03 |
| 872865 | 38.1 | 22.4 | 101620.0 | 0.0208 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 3.4 | 0.0 | 109.7 | 1.2 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 8:02 |
| 872864 | 39.2 | 22.4 | 101620.0 | 0.0209 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 3.4 | 0.0 | 109.7 | 1.2 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 8:01 |
| 872863 | 39.4 | 22.4 | 101620.0 | 0.0209 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 3.4 | 0.0 | 113.3 | 1.2 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 8:00 |
| 872862 | 38.9 | 22.4 | 101620.0 | 0.0212 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 3.4 | 0.0 | 112.8 | 1.2 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 7:59 |
| 872861 | 38.5 | 22.4 | 101620.0 | 0.0213 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 3.4 | 0.0 | 112.5 | 1.3 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 7:58 |
| 872860 | 38.5 | 22.4 | 101620.0 | 0.0214 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 3.4 | 0.0 | 113.7 | 1.4 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 7:57 |
| 872859 | 39.3 | 22.3 | 101620.0 | 0.0214 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 3.4 | 0.0 | 111.8 | 1.4 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 7:56 |
| 872858 | 39.4 | 22.3 | 101620.0 | 0.0214 | 0.0003 | 10.1 | 63.4 | 5.0 | 0.0008 | -0.0001 | 3.4 | 0.0 | 107.6 | 1.5 | 0.0 | 11.10.2023 9:13 | 11.10.2023 9:15 | 25.09.2023 7:55 |

Рис. 8. Таблиця даних станції T3950716

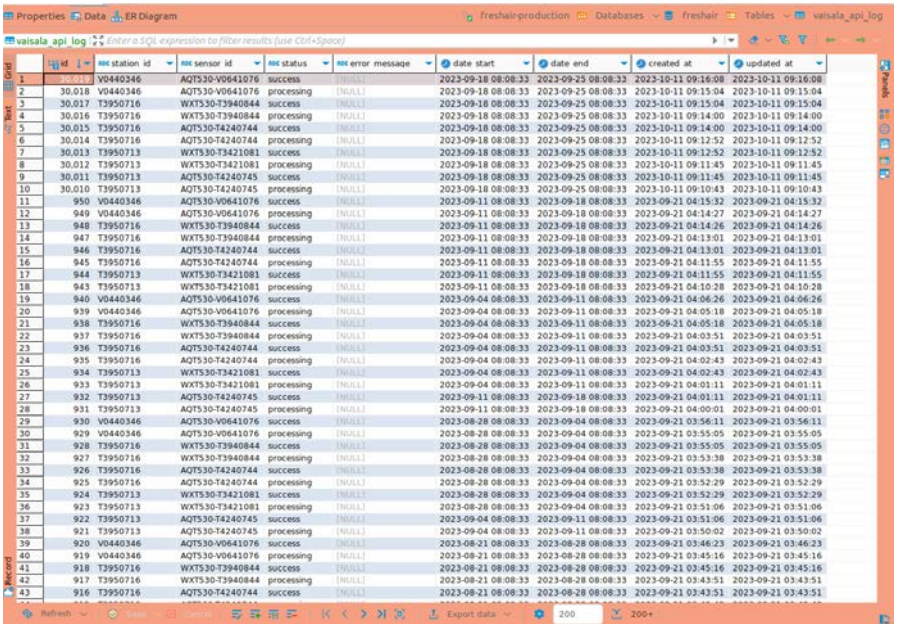

Рис. 9. Таблиця даних vaisala_api_log

Для зручного керування процесом завантаження та ведення статистики оброблених пакетів від станцій Vaisala додано таблицю vaisala_api_log, у якій зберігаються дані про початок і кінець завантаження даних зі станцій.

На віртуальній машині VPS працює операційна система Ubuntu, у якій на основі утиліти cron запускаються дві фонові команди, одна – кожні 10 хвилин, а інша – кожні 20 хвилин. Перша команда завантажує дані зі станцій Vaisala через vaisala-api та зберігає їх у базі даних у таблицях типу station_T3950716. Дані про початок і завершення роботи команди завантаження записуються в таблиці vaisala_api_log. Інша команда перераховує значення з таблиць станцій із дискретизацією в 1 хвилину, де вони перебувають після завантаження, у таблицю vaisala_splits із дискретизацією у 20 хвилин, денною та річною.

Загальну структуру роботи сервера наведено на рис. 10.

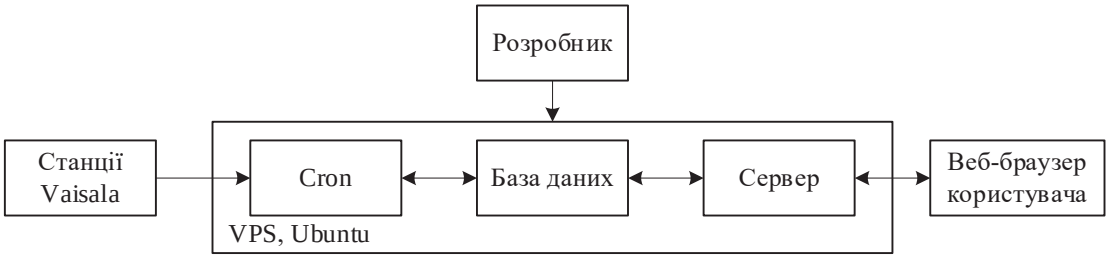


Рис. 10. Структурна схема роботи серверу

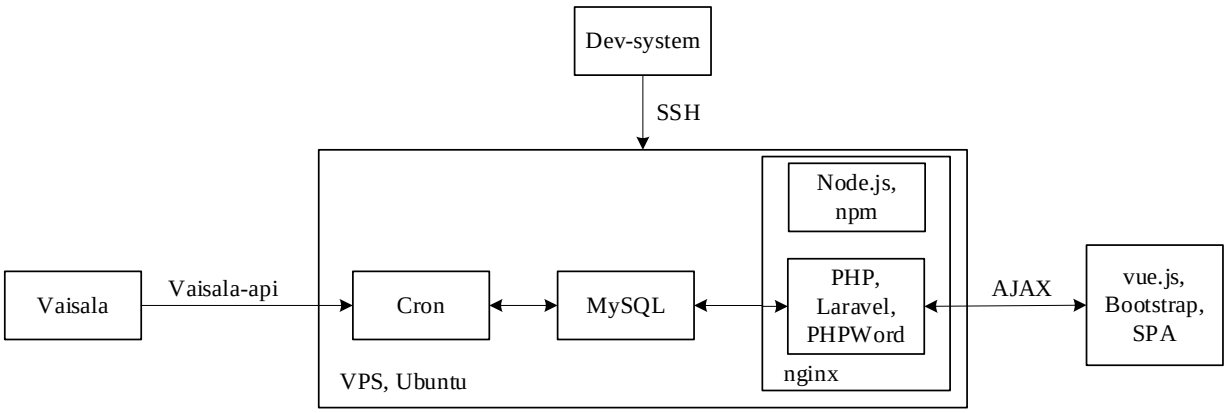


Рис. 11. Структурна схема використаних технологій на сервері

Розробник має змогу отримати доступ до керування сервером за допомогою різних технологій, у тому числі SSH.

SSH (Secure Shell) – це протокол забезпеченого мережевого з'єднання, який дає змогу захищено здійснювати віддалений доступ до комп'ютерів і передавати дані через незахищені мережі, такі як Інтернет. SSH шифрує з'єднання, а також надає автентифікацію та ідентифікацію користувачів, що робить його ідеальним для віддаленого адміністрування серверів і безпечної передачі файлів.

Використання цієї технології дає змогу адмініструвати сервер, оновлювати встановлені бібліотеки та застосунки, працювати з базою даних (для чого потрібне окреме підключення до бази даних і використання клієнта для роботи з нею, яким у цьому випадку є DBeaver). Також за SSH здійснюється керування компіляцією файлів front-end у Node.js та npm. Завдяки компіляції файлів користувацького інтерфейсу можна оптимізувати роботу клієнтської частини інформаційної системи.

Сервер nginx слухає 80 і 443 порти, під час отримання запитів визначається їх тип, якщо запити ініціалізовані, то вони оброблюються Laravel та користувачу надсилається вебсторінка з технологією SPA, яка побудована на основі vue.js та bootstrap з елементами керування та цільовим вмістом, або бінарний DOCX файл, створений PHPWord, якщо запити отримано від не зареєстрованого користувача, то він отримує форму входу в систему. Сторінки оновлюються за технологією AJAX.

SPA (односторінкова програма) – це тип вебдодатка, який взаємодіє з користувачем, завантажуючи лише одну HTML-сторінку під час початкового завантаження та пізніше динамічно оновлюючи вміст сторінки без перезавантаження. Цей підхід зазвичай базується на використанні JavaScript і вебсервісів, які обмінюються даними із сервером.

SPA може покращити швидкодію та реактивність вебдодатків, зменшити потребу в передаванні даних між сервером і клієнтом, а також дозволити створювати більш інтерактивні додатки. Проте важливо правильно управляти маршрутизацією та пам'яттю браузера, щоб уникнути можливих проблем з продуктивністю та SEO.

Код проєкту розміщений на VPS сервері, а також у закритому репозиторії GitHub [20], дані для підключення до якого розміщено в Project credentials.

Загальний код інформаційної системи перебуває в корені проєкту.

Код PHP для cron розміщено в теці app/Console/Commands.

Код у файлі DeleteOldMeasurementData.php видаляє «сирі» дані в таблицях даних станцій. SetVaisala20mSplits.php та інші файли, що містять слово splits, проводять відповідну обробку «сирих» даних, усереднюючи їх за відповідний проміжок, у цьому випадку відбувається усереднення за 20 хвилин, також є денне, місячне усереднення. VaisalaApi.php дає змогу завантажити «сирі» дані зі станцій у базу даних. Завдяки register.php можна додавати нових користувачів із доступом до підключення до створеного сервера.

Код, що розміщений за адресою app/Repositories/Stations та в теці StationsWithRawData, є моделлю рядка таблиці даних, за яким здійснюється обробка «сирих» даних.

Компільовані файли інтерфейсу користувача, що надсилаються у веббраузер користувача, та вихідні матеріали інтерфейсу розміщено в теці frontend.

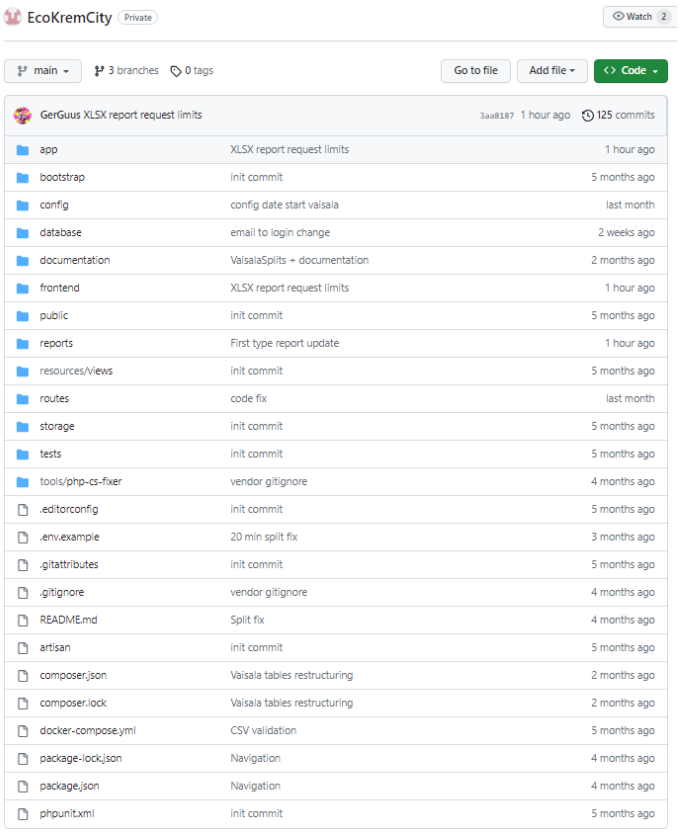


Рис. 12. Вміст проєкту в GitHub

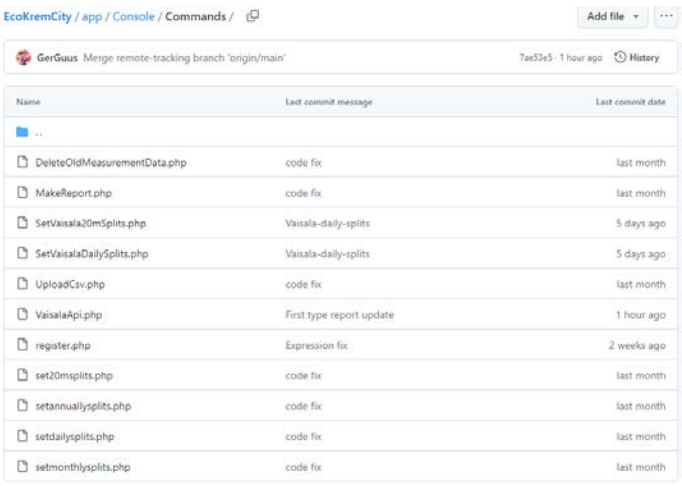


Рис. 13. Файли команд для cron

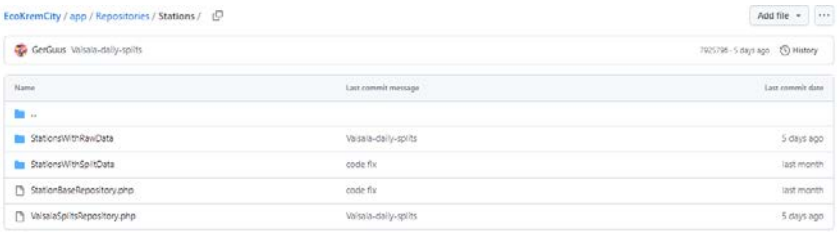


Рис. 14. Код обробки даних

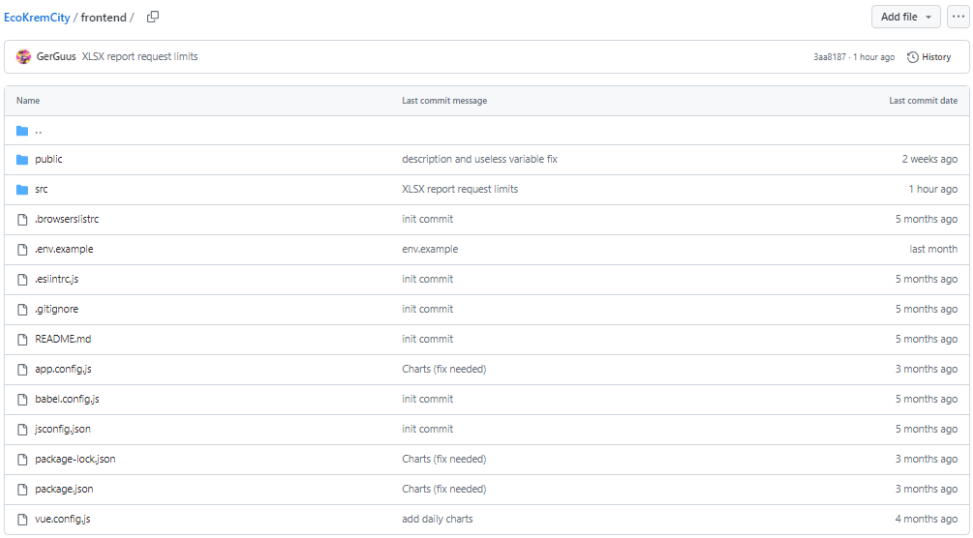


Рис. 15. Матеріали front-end

**Висновки**

Під час виконання дослідження було досягнуто мети з розробки та впровадження інформаційної системи з автоматичного збору, накопичення, структурування даних зі станцій Vaisala й автоматизованого формування звітних відомостей про стан атмосферного повітря.

За результатами аналізу вебресурсів, що надають дані про стан атмосферного повітря та є у відкритому доступі, виявлено, що наразі відсутнє відкрите рішення для автоматизованого формування звітів про якість повітря у текстовому форматі. У деяких ресурсів наявний функціонал для завантаження набору знятих усереднених даних у стандартизованих одинцях виміру чи перерахованих на власні індекси якості повітря.

Мети дослідження досягнуто за допомогою реалізації інформаційної системи, що базується на ОС Ubuntu, сервері Nginx, системі керування базами даних MySQL, мові PHP з фреймворками PHPWord та Laravel, бібліотеками JavaScript Vue.js, Bootstrap і мережевих технологіях vaisala_api, SPA, AJAX. Сервер модифіковано для обробки великого об'єму даних зі станцій Vaisala, для яких створені окремі таблиці для вивантаження «сирих» даних і таблиця для перерахунку хвилинних показів у 20-хвилинні, денні, місячні та річні.

Наукова новизна роботи полягає в тому, що вперше створено метод автоматизованого генерування звітів, що базується на збиранні, обробці та записі даних від станцій Vaisala у базу даних SQL за допомогою cron-команд через Vaisala API.

У подальших дослідженнях буде розроблено метод формування звітів про кількість перевищень встановлених нормативів маркерів атмосферного повітря за постановою №827, що дасть змогу інтегрувати його в розроблену інформаційну систему та підвищити автоматизацію звітування в комунальному підприємстві.

***Література***

1. DOAJ. Directory Of Open Access Journals. URL: https://doaj.org/ (дата звернення: 23.10.2023).
2. ScienceDirect. Search for peer-reviewed journal articles and book chapters (including open access content). URL: https://www.sciencedirect.com/ (дата звернення: 23.10.2023).
3. Crossref. Search the metadata of journal articles, books, standards, datasets & more. URL: https://search.crossref.org/ (дата звернення: 23.10.2023).
4. Google Scholar. URL: https://scholar.google.com/ (дата звернення: 23.10.2023).
5. DuckDuckGo. URL: https://duckduckgo.com/ (дата звернення: 23.10.2023).
6. RisÃ¸, Denmark Air Pollution: Real-time Air Quality Index. aqicn.org. URL: https://aqicn.org/city/denmark/riso/ (дата звернення: 23.10.2023).
7. The World's Most Accurate Air Quality Data – BreezoMeter. Custom Map Tools & Products – Google Maps Platform. URL: https://www.breezometer.com/air-quality-map/pollen (дата звернення: 23.10.2023).
8. AirNow.gov. URL: https://www.airnow.gov/?city=Washington&state=DC&country=USA (дата звернення: 23.10.2023).
9. Washington, DC Air Quality Index | AccuWeather. URL: https://www.accuweather.com/en/us/washington/20006/air-quality-index/327659 (дата звернення: 23.10.2023).
10. High-performance, Cheap, and Stable VDS/VPS Powered by KVM and SSD NVMe. Zomro.com. URL: https://zomro.com/vds (дата звернення: 23.10.2023).

11. nginx news. URL: https://nginx.org/ (дата звернення: 23.10.2023).
12. MySQL. URL: https://www.mysql.com/ (дата звернення: 23.10.2023).
13. PHP: Hypertext Preprocessor. URL: https://www.php.net/ (дата звернення: 23.10.2023).
14. Laravel – The PHP Framework For Web Artisans. URL: https://laravel.com/ (дата звернення: 23.10.2023).
15. PHPWord. GitHub Pages. URL: https://phpoffice.github.io/PHPWord/ (дата звернення: 23.10.2023).
16. DBeaver Community | Free Universal Database Tool. URL: https://dbeaver.io/ (дата звернення: 23.10.2023).
17. Node.js. URL: https://nodejs.org (дата звернення: 23.10.2023).
18. Vue.js – The Progressive JavaScript Framework. URL: https://vuejs.org/ (дата звернення: 23.10.2023).
19. Bootstrap. Bootstrap The most popular HTML, CSS, and JS library in the world. URL: https://getbootstrap.com/ (дата звернення: 23.10.2023).
20. EcoKremCity. URL: https://github.com/EcoKremCity/EcoKremCity (дата звернення: 23.10.2023).

### *References*

1. *DOAJ. Directory Of Open Access Journals.* Retrieved from: https://doaj.org/ [in English].
2. *ScienceDirect. Search for peer-reviewed journal articles and book chapters (including open access content).* Retrieved from: https://www.sciencedirect.com/ [in English].
3. *Crossref. Search the metadata of journal articles, books, standards, datasets & more.* Retrieved from: https://search.crossref.org/ [in English].
4. *Google Scholar.* Retrieved from: https://scholar.google.com/ [in English].
5. *DuckDuckGo.* Retrieved from: https://duckduckgo.com/ [in English].
6. *RisÃ¸ Denmark Air Pollution: Real-time Air Quality Index. aqicn.org.* Retrieved from: https://aqicn.org/city/denmark/riso/ [in English].
7. *The World's Most Accurate Air Quality Data – BreezoMeter. Custom Map Tools & Products – Google Maps Platform.* Retrieved from: https://www.breezometer.com/air-quality-map/pollen [in English].
8. *AirNow.gov.* Retrieved from: https://www.airnow.gov/?city=Washington&state=DC&country=USA [in English].
9. *Washington, DC Air Quality Index | AccuWeather.* Retrieved from: https://www.accuweather.com/en/us/washington/20006/air-quality-index/327659 [in English].
10. *High-performance, Cheap, and Stable VDS/VPS Powered by KVM and SSD NVMe. Zomro.com.* Retrieved from: https://zomro.com/vds [in English].
11. *nginx news.* Retrieved from: https://nginx.org/ [in English].
12. *MySQL.* Retrieved from: https://www.mysql.com/ [in English].
13. *PHP: Hypertext Preprocessor.* Retrieved from: https://www.php.net/ [in English].
14. *Laravel – The PHP Framework For Web Artisans.* Retrieved from: https://laravel.com/ [in English].
15. *PHPWord. GitHub Pages.* Retrieved from: https://phpoffice.github.io/PHPWord/ [in English].
16. *DBeaver Community | Free Universal Database Tool.* Retrieved from: https://dbeaver.io/ [in English].
17. *Node.js.* Retrieved from: https://nodejs.org [in English].
18. *Vue.js – The Progressive JavaScript Framework.* Retrieved from: https://vuejs.org/ [in English].
19. *Bootstrap. Bootstrap The most popular HTML, CSS, and JS library in the world.* Retrieved from: https://getbootstrap.com/ [in English].
20. *EcoKremCity.* Retrieved from: https://github.com/EcoKremCity/EcoKremCity [in English].

**УДК 004.021**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-05**

# РОЗРОБКА МЕТОДУ АВТОМАТИЧНОГО ФОРМУВАННЯ ЗВІТНОСТІ ПРО КІЛЬКІСТЬ ПЕРЕВИЩЕНЬ ВСТАНОВЛЕНИХ НОРМАТИВІВ МАРКЕРІВ АТМОСФЕРНОГО ПОВІТРЯ

***Вадурін К. О.,*** *аспірант, Кременчуцький національний університет імені Михайла Остроградського, Кременчук, Україна*
*https://orcid.org/0000-0001-7781-5783, kir3337@gmail.com*

***Перекрест А. Л.,*** *д-р техн. наук, професор, Кременчуцький національний університет імені Михайла Остроградського, Кременчук, Україна*
*https://orcid.org/0000-0002-7728-9020, pksg13@gmail.com*

***Бахарєв В. С.,*** *д-р техн. наук, професор, Кременчуцький національний університет імені Михайла Остроградського, Кременчук, Україна*
*https://orcid.org/0000-0001-9312-654X, v.s.baharev@gmail.com*

***Анотація.*** *В Україні наразі спостерігаються значні суспільні зміни, які охоплюють цифровізацію різних аспектів соціального й наукового життя. Зокрема, впроваджуються цифрові технології у галузь екологічного моніторингу, де діє комунальне підприємство «Науковий центр еколого-соціальних досліджень» при Кременчуцькій міській раді в Полтавській області. Одним з основних завдань для цього підприємства є стандартизація внутрішнього документообігу для відповідності новим стандартам у проведенні екологічних досліджень. У цьому контексті важливо розробити метод автоматичного формування звітів про кількість перевищень встановлених нормативів маркерів атмосферного повітря, завдяки чому можна скоротити час, який оператор витрачає на підготовку звітності. Основною метою цього дослідження є створення методу автоматичного формування звітів про кількість перевищень маркерами нормативів атмосферного повітря відповідно до постанови № 827 КМУ. Для реалізації поставленої мети спочатку було проаналізовано ключові тези постанови й визначено основні положення, яких слід дотримуватися під час формування звіту. Потім, за витримками з додатків до постанови, в Excel було згенеровано набір даних, що містить усі визначені маркери, а також нормативи для подальшого тестування розробленого методу. За згенерованими даними побудовано ряд таблиць усереднення: годинну; восьмигодинну; добову; добову за максимальним восьмигодинним значенням; місячну; річну. За таблицями усереднення та пунктами Додатку 2 до Порядку здійснення державного моніторингу в галузі охорони атмосферного повітря загальний звіт розділено на три аркуші Excel, у яких реалізовано функціонал підрахунку кількості перевищень маркерами атмосферного повітря встановлених нормативів. За результатами виконання дослідження розроблено метод, що базується на основі використання ковзкої області виділення комірок і припущення, що дата сет починається з 01.01.2020 0:00:00.*

***Ключові слова:*** *якість атмосферного повітря, екологічний моніторинг, Excel, метод автоматичного розрахунку.*

# DEVELOPMENT OF A METHOD OF AUTOMATIC REPORTING ON THE NUMBER OF EXCEEDANCES OF THE ESTABLISHED STANDARDS OF ATMOSPHERIC AIR MARKERS

***Kyrylo Vadurin,*** *Postgraduate Student, Kremenchuk Mykhailo Ostrohradskyi National University, Kremenchuk, Ukraine*
*https://orcid.org/0000-0001-7781-5783, kir3337@gmail.com*

***Andrii Perekrest,*** *Dc.S., Prof., Kremenchuk Mykhailo Ostrohradskyi National University, Kremenchuk, Ukraine*
*https://orcid.org/0000-0002-7728-9020, pksg13@gmail.com*

***Volodymyr Bakharev,*** *Dc.S., Prof., Kremenchuk Mykhailo Ostrohradskyi National University, Kremenchuk, Ukraine*
*https://orcid.org/0000-0001-9312-654X, v.s.baharev@gmail.com*

***Abstract.*** *Ukraine is currently experiencing significant social changes that include the digitalisation of various aspects of social and scientific life. In particular, digital technologies are being introduced in the field of environmental monitoring, where the Municipal Enterprise "Scientific Centre for Environmental and Social*

*Research" operates under the Kremenchuk City Council in Poltava Oblast. One of the main challenges for this enterprise is to standardise its internal document flow to meet new standards in environmental research. In this context, it is important to develop a method for automatically generating reports on the number of exceedances of the established standards of atmospheric air markers, which will reduce the time spent by the operator on reporting. The main purpose of this study is to create a method for automatically generating reports on the number of exceedances of air quality standards by markers in accordance with Resolution No. 827 of the Cabinet of Ministers of Ukraine. To achieve this goal, we first analysed the key points of the resolution and identified the main provisions that should be followed when generating the report. Then, based on the excerpts from the annexes to the resolution, a data set containing all the identified markers and standards was generated in Excel for further testing of the developed method. Based on the generated data, a number of averaging tables were built: hourly; eight-hourly; daily; daily by the maximum eight-hour value; monthly; and annual. According to the averaging tables and the paragraphs of Annex 2 to the Procedure for State Monitoring in the Field of Atmospheric Air Protection, the general report is divided into three Excel sheets, which implement the functionality of calculating the number of exceedances of the established standards by atmospheric air markers. As a result of the study, a method was developed based on the use of a sliding cell selection area and the assumption that the set date starts from 01.01.2020 0:00:00.*

***Key words:*** *air quality; environmental monitoring; Excel; automatic calculation method.*

**Вступ**

Наразі в Україні відбуваються значні суспільні зміни – проводиться цифровізація всіх процесів соціальної та наукової діяльності: вводяться ID-картки замість паспортів; розвивається цифрова платформа індивідуального документообігу «ДІЯ»; у лікарнях запроваджуються цифрові лікарняні картки. Інтеграція цифрових технологій не оминула і сферу екологічного моніторингу, у якій працює комунальне підприємство «Науковий центр еколого-соціальних досліджень» Кременчуцької міської ради Кременчуцького району Полтавської області. Наразі однією з основних завдань підприємства є узгодження внутрішнього документообігу з оновленими стандартами провадження екологічних досліджень. Тому актуальною задачею є розробка методу автоматичного формування звітності про кількість перевищень встановлених нормативів маркерів атмосферного повітря, завдяки чому можна зменшити час оператора на підготовку повного комплекту звітності й попередити помилки в розрахунках, викликані людським фактором.

**Метою дослідження** є розробка методу автоматичного формування звітності про кількість перевищень встановлених нормативів маркерів атмосферного повітря згідно з постановою № 827 КМУ [1], що дасть змогу узгодити внутрішній документообіг зі встановленими стандартами й удосконалити створену інформаційну систему з формування звітних відомостей.

**Виконання досліджень**

Спочатку було проведено пошук існуючих науково-технічних рішень з автоматичного формування звітності про кількість перевищень встановлених нормативів маркерів атмосферного повітря згідно з постановою № 827 КМУ за наукометричними базами [2–4] та доступними через стандартні пошукові системи [5–6]. Під час пошуку не виявлено подібних рішень.

Тому на початковому етапі дослідження було проаналізовано основні положення постанови Кабінету Міністрів України № 827 від 14 серпня 2019 року «Деякі питання здійснення державного моніторингу в галузі охорони атмосферного повітря». Основні положення, наведені в постанові, такі:

1. Затверджено Порядок здійснення державного моніторингу в галузі охорони атмосферного повітря.
2. Внесено зміни до Положення про державну систему моніторингу довкілля.
3. Визнано такими, що втратили чинність, низку постанов Кабінету Міністрів України з питань моніторингу атмосферного повітря.
4. Доручено обласним держадміністраціям та іншим органам виконавчої влади визначити органи управління якістю атмосферного повітря, утворити відповідні комісії, розробити програми моніторингу тощо.
5. Встановлено, що Порядок набирає чинності через рік після опублікування постанови, а інші пункти – з дня опублікування.

Основні положення, яких слід притримуватися під час формування звіту на основі цієї постанови:

1. Звіт має містити узагальнені дані про якість атмосферного повітря за певний проміжок часу на певній території.
2. У звіті має бути наведена оцінка стану атмосферного повітря й атмосферних опадів.
3. Звіт повинен містити прогнози стану атмосферного повітря та його змін.
4. У звіті надається інформація про вплив рівнів забруднювальних речовин в атмосферному повітрі на здоров'я та життєдіяльність населення.
5. У звіті зазначаються перевищення граничних величин, цільових показників і порогів небезпеки.
6. Звіт має містити рекомендації щодо заходів для поліпшення якості атмосферного повітря.
7. Дані у звіті мають бути обґрунтовані посиланнями на результати вимірювань, моделювання та розрахунків.

8. Висновки звіту мають ґрунтуватися на чинних нормативно-правових актах і науково обґрунтованих методиках.

9. Звіт має бути оформлений згідно з вимогами нормативних документів і стандартів.

Під час попередніх досліджень було отримано певний масив даних зі станцій Vaisala, але не всі зі встановлених у постанові маркерів вимірюються станціями. Тому для розробки методу формування звітів за постановою було згенеровано набір даних, що містить усі визначені маркери, а також стовпці для подальшого прогнозування змін атмосферного повітря.

Розробка методу проводилася в Microsoft Excel – це потужний процесор для обробки даних та електронних таблиць, розроблений корпорацією Microsoft. Програма надає зручне інтерфейсне середовище для створення, редагування, організації та аналізу числових даних. Excel може використовуватися для різних завдань, включно зі створенням фінансових звітів, графіків, таблиць, обчислень, аналізом статистики тощо.

Умовністю генерованого списку та подальших обчислень є те, що дані мають починатися 1 січня будь-якого року о 00:00:00 (від 1900 року до 9999 року, що є системним обмеженням Excel).

Генеровані дані зібрані в «розумну» таблицю, що дає змогу автоматично заповнювати всі її стовпці, але викликає обмеження з адресуванням на комірки даних – якщо формула посилатиметься на комірки поза таблицею, то розрахунки проводитися не будуть.

Стовпець «Дата та час» згенерований за допомогою написання двох послідовних інтервалів часу, і «розумна» таблиця продовжує написання дати та часу з тим самим інтервалом.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Дата та час | Вологість, % RH | Тиск, гПа | Температура, °C | Діоксид сірки, мкг/куб. м | Діоксид азоту та оксиди азоту | Бензол, мкг/куб. м | Оксид вуглецю, мг/куб. м | Свинець, мкг/куб. м | Тверді частки 10, мкг/куб. м | Тверді частки 2,5, мкг/куб. | Арсен, нг/куб. м | Кадмій, нг/куб. м | Нікель, нг/куб. м | Бенз(а)пірен, нг/куб. м | Озон, мкг/куб. | Напрямок вітру |
| 2 | 01.01.20 0:00 | 87,00 | 1019,75 | -4,00 | 36,752 | 23,200 | 1,113 | 2,150 | 6,030 | 48,820 | 15,990 | 1,750 | 0,500 | 8,750 | 0,588 | 72,070 | 135 |
| 3 | 01.01.2020 | 93,00 | 1019,91 | -4,00 | 50,971 | 17,440 | 2,127 | 9,910 | 7,950 | 8,750 | 6,200 | 2,150 | 0,430 | 12,060 | 0,280 | 25,600 | 90 |
| 4 | 01.01.20 0:40 | 85,00 | 1019,33 | -4,00 | 20,369 | 10,480 | 2,902 | 5,730 | 6,530 | 23,260 | 5,700 | 2,130 | 1,290 | 9,480 | 0,377 | 72,150 | 90 |
| 5 | 01.01.2020 | 95,00 | 1019,12 | -5,00 | 5,117 | 28,200 | 2,300 | 10,110 | 0,970 | 14,130 | 3,880 | 0,160 | 2,490 | 0,310 | 0,040 | 107,440 | 90 |
| 6 | 01.01.20 1:20 | 94,00 | 1019,09 | -5,00 | 4,050 | 10,970 | 10,000 | 9,380 | 7,150 | 6,360 | 9,810 | 1,140 | 4,610 | 3,420 | 0,141 | 97,800 | 135 |
| 7 | 01.01.2020 | 91,00 | 1019,06 | -4,00 | 11,982 | 31,880 | 1,328 | 4,670 | 1,110 | 5,970 | 9,150 | 0,280 | 3,510 | 65,880 | 0,489 | 20,180 | 315 |
| 8 | 01.01.20 2:00 | 80,00 | 1019,67 | -4,00 | 32,735 | 29,630 | 1,543 | 2,400 | 7,150 | 23,640 | 3,340 | 3,040 | 2,190 | 1,690 | 0,088 | 22,220 | 90 |
| 9 | 01.01.2020 | 83,00 | 1019,13 | -4,00 | 19,375 | 27,140 | 2,066 | 7,120 | 7,570 | 10,980 | 6,470 | 0,980 | 0,100 | 1,230 | 0,051 | 70,490 | 90 |
| 10 | 01.01.20 2:40 | 91,00 | 1019,12 | -4,00 | 47,104 | 12,110 | 1,008 | 9,330 | 3,280 | 24,940 | 9,870 | 1,930 | 3,360 | 0,750 | 0,475 | 104,210 | 225 |
| 11 | 01.01.2020 | 88,00 | 1019,48 | -4,00 | 37,376 | 21,820 | 0,331 | 4,050 | 0,390 | 21,480 | 2,900 | 3,070 | 1,050 | 11,450 | 0,170 | 35,140 | 315 |
| 12 | 01.01.20 3:20 | 84,00 | 1019,33 | -4,00 | 45,318 | 22,200 | 2,909 | 8,010 | 7,010 | 21,390 | 14,310 | 4,800 | 0,630 | 4,270 | 0,241 | 72,530 | -1 |

Рис. 1. Згенерована таблиця даних

Стовпець «Вологість, % RH» згенерований з використанням функції:

=SWITCH(MONTH($A2);1;RANDBETWEEN(80;95);2;RANDBETWEEN(80;95);12;RANDBETWEEN(80;95);6;RANDBETWEEN(45;51);7;RANDBETWEEN(45;51);8;RANDBETWEEN(45;51);3;RANDBETWEEN(60;70);4;RANDBETWEEN(60;70);5;RANDBETWEEN(60;70);9;RANDBETWEEN(60;70);10;RANDBETWEEN(60;70);11;RANDBETWEEN(60;70)).

Функція генерації вологості дає змогу врахувати сезонність [32], але не враховує час дня. Загальна функція базується на даних першого стовпця, де введена дата. Функція MONTH($A2) отримує номер місяця, і залежно від отриманого значення відбувається переключення умов генерації вологості у функції SWITCH(). Генерація здійснюється за допомогою функції RANDBETWEEN(80;95). Для кожного сезону вологість коливається в межах 10–15% від статистичної.

Атмосферний тиск для стовпця «Тиск, гПа» згенеровано на рівні 1019–020 гПа функцією:

=RAND()*100/100+1019.

Для генерації тиску використовується функція RAND(), що повертає випадкове число від 0 до 1.

Стовпець «Температура, °C» згенеровано сезонно, подібно до вологості функцією:

=SWITCH(MONTH($A2);1;RANDBETWEEN(-5;-4);2;RANDBETWEEN(-4;-3);3;RANDBETWEEN(1;2);4;RANDBETWEEN(8;10);5;RANDBETWEEN(14;16);6;RANDBETWEEN(18;19);7;RANDBETWEEN(19;20);8;RANDBETWEEN(18;19);9;RANDBETWEEN(13;14);10;RANDBETWEEN(7;8);11;RANDBETWEEN(1;2);12;RANDBETWEEN(-3;-1)).

Генерування маркерів атмосферного повітря проводилося на підставі даних з Додатку 2 до Порядку, визначених у пункті «II. Верхній та нижній пороги оцінювання», щоб значення балансували на межі допустимих рівнів та інколи перевищували їх, для відображення відповідної статистики у звіті. Вірогідність значного перевищення гранично допустимих концентрацій під час генерації враховується за допомогою функції IF(), яка очікує випадкове число, що менше за певне значення (RAND()>0,1).

Формули для генерування деяких із них:

– Діоксид сірки, мкг/куб. м:

=IF(RAND()>0,1;RANDBETWEEN(0;51000)/1000;
RANDBETWEEN(3000;80000)/100);

– Діоксид азоту й оксиди азоту, мкг/куб. м:

=IF(RAND()>0,05;RANDBETWEEN(1000;3200)/100;
RANDBETWEEN(100;8000)/100);

– Бензол, мкг/куб. м:

=IF(RAND()>0,05;RANDBETWEEN(10;3000)/1000;
RANDBETWEEN(5;10)).

Також у таблиці згенеровано напрямки вітру в числовому представленні кута руху повітряних мас. Значення напрямків вітру відносно назв можливих напрямків вітру: північний – 1; північно-східний – 45; східний – 90; південно-східний – 135; південний – 180; південно-західний – 225; західний – 270; північно-західний – 315; штиль – –1.

Функція для генерації напрямку вітра спочатку генерує випадкове ціле число від 0 включно до 9 не включно за допомогою INT(RAND()*9, і далі встановлюється випадковий напрямок функцією SWITCH():

=SWITCH(INT(RAND()*9);0;1;1;45;2;90;3;135;4;180;5;225;6;270;7;315;8;-1).

Для усереднення даних за годину спочатку було сформовано стовпець «Дата та час». Цей стовпець фактично генерується у вигляді посилань на стовпець таблиці з 20-хвилинним усередненням і бере звідти дані лише для конкретної години. Функція для отримання години має такий вигляд:

=INDEX(Таблиця1[Дата та час];(ROW(Таблиця1[@[Дата та час]])-1)*3+1;0).

Функція працює так: функція ROW(Таблиця1[@[Дата та час]])-1)*3+1 для комірки з другого рядка повертає номер рядка другої комірки вихідної таблиці, для третьої комірки повертає 8-му комірку вихідної таблиці, для четвертої – 11-ту і так далі. А функція INDEX() дає змогу отримати значення цієї комірки.

Усереднення даних в усіх інших стовпцях, крім напрямку вітру, реалізується такою функцією:

=AVERAGE(OFFSET(Таблиця1[[#Заголовки];[Вологість,

% RH]];IF(ROW([@[Дата та час]])>2;(ROW([@[Дата та час]])-2)*3+2;1);0;IF(ROW([@[Дата та час]])=2;4;3);1))

Оскільки виходимо з того, що дані починаються 01.01.2020 з 00:00, то перше усереднення буде мати 4 виміри: 00:00, 00:20, 00:40, 01:00, а всі інші усереднення матимуть 3 виміри: 01:20, 01:40, 02:00, використаємо функцію IF(), яка визначатиме номер рядка, до якого застосовується формула, і якщо рядок не перший, то відбуватиметься зміщення виділення за допомогою OFFSET() на наступні 3 комірки, якщо ж рядок перший, то виділятимуться 4 перші комірки. У функції OFFSET() «Таблиця1[[#Заголовки];[Вологість,% RH]]» вказує на стовпець, до якого застосовується ковзке зміщення. Функція AVERAGE() дає змогу знайти середнє значення комірок, виділених за допомогою ковзкого зміщення.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Дата та час | Вологість, % RH | Тиск, гПа | Температура, °C | Діоксид сірки, мкг/куб. м | Діоксид азоту та оксиди азоту, мкг/куб. м | Бензол, мкг/куб. м | Оксид вуглецю, мг/куб. м | Свинець, мкг/куб. м | Тверді частки 10, мкг/куб. м | Тверді частки 2,5, мкг/куб. м | Арсен, нг/куб. м | Кадмій, нг/куб. м | Нікель, нг/куб. м | Бенз(а)пірен, нг/куб. м | Озон, мкг/куб. | Напрямок вітру | Допоміжний стовпець для розрахунку перевищень діоксиду сірки ВНП827ч4 | Допоміжний стовпець для розрахунку перевищень діоксиду азоту ВНП827ч4 | Допоміжний стовпець для розрахунку перевищень озону ВНП827ч4 | Допоміжний стовпець для розрахунку перевищень озону-інф ВНП827ч4 |
| 2 | 01.01.20 1:00 | 85,75 | 1019,65 | -4,75 | 168,83 | 41,74 | 1,00 | 6,80 | 11,29 | 17,43 | 16,14 | 3,30 | 0,98 | 17,08 | 0,23 | 48,36 | 315 | 0,00 | 0,00 | 0,00 | 0,00 |
| 3 | 01.01.20 2:00 | 87,67 | 1019,56 | -4,33 | 179,87 | 31,40 | 1,51 | 9,08 | 4,06 | 14,80 | 11,12 | 2,05 | 1,43 | 9,54 | 0,34 | 42,22 | 180 | 0,00 | 0,00 | 0,00 | 0,00 |
| 4 | 01.01.20 3:00 | 84,67 | 1019,51 | -4,33 | 241,26 | 22,27 | 1,74 | 4,55 | 4,28 | 11,03 | 8,45 | 1,99 | 1,50 | 9,52 | 0,32 | 63,69 | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 5 | 01.01.20 4:00 | 91,67 | 1019,42 | -4,67 | 23,41 | 20,46 | 1,63 | 5,07 | 2,27 | 16,85 | 7,62 | 1,50 | 2,83 | 8,92 | 0,28 | 97,20 | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 6 | 01.01.20 5:00 | 84,33 | 1019,72 | -4,67 | 74,69 | 25,72 | 1,58 | 9,45 | 4,77 | 12,36 | 16,90 | 1,46 | 2,40 | 7,29 | 0,43 | 49,88 | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 7 | 01.01.20 6:00 | 85,67 | 1019,52 | -4,33 | 260,04 | 29,24 | 0,74 | #### | 3,85 | 21,34 | 8,54 | 4,88 | 1,81 | 37,77 | 0,12 | 55,27 | 225 | 0,00 | 0,00 | 0,00 | 0,00 |
| 8 | 01.01.20 7:00 | 86,67 | 1019,56 | -4,33 | 7,47 | 21,74 | 1,27 | 4,45 | 1,95 | 15,26 | 12,27 | 3,03 | 1,76 | 4,88 | 0,18 | 67,58 | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 9 | 01.01.20 8:00 | 86,33 | 1019,69 | -4,33 | 144,70 | 21,60 | 1,76 | 4,46 | 3,94 | 16,13 | 13,55 | 1,29 | 1,73 | 9,35 | 0,41 | 58,99 | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 10 | 01.01.20 9:00 | 94,33 | 1019,59 | -4,67 | 30,09 | 26,14 | 2,00 | 7,53 | 3,28 | 19,73 | 9,91 | 2,09 | 2,14 | 4,91 | 0,23 | 60,15 | 90 | 0,00 | 0,00 | 0,00 | 0,00 |
| 11 | 01.01.20 10:00 | 90,00 | 1019,30 | -4,33 | 16,18 | 23,24 | 1,80 | 9,07 | 4,82 | 32,15 | 17,94 | 1,78 | 1,25 | 5,94 | 0,42 | 47,09 | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 12 | 01.01.20 11:00 | 84,67 | 1019,35 | -4,33 | 24,79 | 18,77 | 1,08 | 7,05 | 5,38 | 20,07 | 8,87 | 2,16 | 2,28 | 10,64 | 0,16 | 81,64 | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 13 | 01.01.20 12:00 | 86,00 | 1019,53 | -4,33 | 28,98 | 28,23 | 3,79 | 8,73 | 5,46 | 15,72 | 15,45 | 3,78 | 2,15 | 34,46 | 0,27 | ##### | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 14 | 01.01.20 13:00 | 90,67 | 1019,41 | -4,33 | 211,98 | 19,68 | 1,53 | 3,90 | 3,73 | 22,00 | 5,45 | 1,31 | 1,49 | 20,71 | 0,21 | ##### | #Н/Д | 0,00 | 0,00 | 0,00 | 0,00 |
| 15 | 01.01.20 14:00 | 84,67 | 1019,46 | -4,33 | 35,40 | 17,47 | 0,38 | 5,60 | 6,36 | 15,33 | 9,41 | 1,74 | 3,12 | 6,54 | 0,37 | 68,94 | 135 | 0,00 | 0,00 | 0,00 | 0,00 |
| 16 | 01.01.20 15:00 | 88,33 | 1019,66 | -4,00 | 33,25 | 20,50 | 1,58 | 6,61 | 4,26 | 16,14 | 10,20 | 1,42 | 2,87 | 10,92 | 0,42 | 62,74 | 315 | 0,00 | 0,00 | 0,00 | 0,00 |

Рис. 2. Таблиця усереднення за годину

Визначення переважаючого напрямку вітру здійснюється подібною функцією, як і підрахунок середнього, але замість AVERAGE() використовується MODE.SNGL(), яка може визначати моду чисел в ряді, саме тому прийнято рішення записувати напрямок вітру не текстом, а у вигляді кутів. Функція визначення переважаючого напрямку вітру має вигляд:

=MODE.SNGL(OFFSET(Таблиця1[[#Заголовки];[Напрямок вітру]];IF(ROW([@[Дата та час]])>2;(ROW([@[Дата та час]])-2)*3+2;1);0;IF(ROW([@[Дата та час]])=2;4;3);1)).

Якщо функції не вдається визначити моду, то повертається значення #Н/Д.

Також до таблиці додано 4 допоміжні рядки для розрахунку критеріїв, вказаних у постанові № 827, щодо визначення кількостей послідовного перевищення показників діоксиду сірки, діоксиду азоту та двох маркерів озону протягом трьох послідовних годин.

Для виявлення зазначених перевищень 2 перші рядки, що відповідають за час 01:00, 02:00, заповнено нулями.

Розрахунок починається з третьої комірки функцією:

=IFS(OR(R2=1;R3=1);0;OR(E2<ВНП827ч4!$G$21;E3<ВНП827ч4!$G$21;[@[Діоксид сірки, мкг/куб. м]]<ВНП827ч4!$G$21);0;AND(E2>=ВНП827ч4!$G$21;E3>=ВНП827ч4!$G$21;[@[Діоксид сірки, мкг/куб. м]]>=ВНП827ч4!$G$21);1).

Тут для визначення перевищень використовується логіка подвійної перевірки функцією IFS(), яка спочатку перевіряє, чи наявні одиниці у двох попередніх комірках, у цьому ж стовпці, що дає змогу запобігти подвійній індикації, яка може виникнути, якщо перевищення спостерігається протягом 4 чи 5 годин, якщо 1 не виявлено, то відбувається перехід до перевірки наступної умови, яка передбачає аналіз 3 послідовних комірок у цільовому стовпці за час 01:00, 02:00, 03:00. У наступній комірці у другій умові відбувається перевірка годин 02:00, 03:00, 04:00 і так далі.

Таблиця усереднення за 8 годин виконана за даними з таблиці 20-хвилинного усереднення за допомогою функцій, які розроблені для годинного усереднення, але з урахуванням збільшення розміру ковзкого вікна.

Для визначення часу восьмигодинного усереднення функцію для стовпця «Дата та час» було змінено таким чином:

=INDEX(Таблиця1[Дата та час];(ROW(Таблиця1[@[Дата та час]])-1)*24+1;0).

Відмінністю між функціями є коефіцієнт 24, що відповідає за кількість комірок, на які відбувається крок.

Для стовпців, у заповненні яких бере участь ковзка область виділення, функції змінено таким чином:

=AVERAGE(OFFSET(Таблиця1[[#Заголовки];[Тиск,
гПа]];IF(ROW([@[Дата та час]])>2;(ROW([@[Дата та час]])-2)*24+2;1);0;IF(ROW([@[Дата та час]])>2;24;25);1)).

У цій формулі зміни також стосуються лише розміру ковзкої області виділення та кроку у 24 комірки. А перша комірка в таблиці матиме область підрахунку у 25 комірок.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Дата та час | Вологість, % RH | Тиск, гПа | Температура, °C | Діоксид сірки, мкг/куб. м | Діоксид азоту та оксиди азоту, мкг/куб. м | Бензол, мкг/куб. м | Оксид вуглецю, мг/куб. м | Свинець, мкг/куб. м | Тверді частки 10, мкг/куб. м | Тверді частки 2,5, мкг/куб. м | Арсен, нг/куб. м | Кадмій, нг/куб. м | Нікель, нг/куб. м | Бенз(а)пірен, нг/куб. м | Озон, мкг/куб. | Напрямок вітру |
| 2 | 01.01.20 0:00 | 81,00 | 1019,69 | -4,00 | 37,182 | 30,470 | 1,514 | 2,120 | 5,960 | 19,200 | 14,980 | 2,780 | 0,740 | 11,360 | 0,001 | 0,610 | 1 |
| 3 | 01.01.2020 0:20 | 90,00 | 1019,33 | -5,00 | 61,690 | 31,930 | 2,260 | 6,410 | 0,910 | 23,950 | 26,910 | 1,080 | 1,390 | 8,840 | 0,041 | 87,760 | 135 |
| 4 | 01.01.20 0:40 | 82,00 | 1019,36 | -4,00 | 37,069 | 10,700 | 0,944 | 7,890 | 7,460 | 25,730 | 5,580 | 1,980 | 0,640 | 5,220 | 0,117 | 63,410 | 45 |
| 5 | 01.01.2020 1:00 | 85,00 | 1019,51 | -5,00 | 1,809 | 25,300 | 2,805 | 2,750 | 4,680 | 22,210 | 13,180 | 0,690 | 3,100 | 6,810 | 0,477 | 106,260 | 180 |
| 6 | 01.01.20 1:20 | 90,00 | 1019,83 | -4,00 | 18,362 | 13,420 | 1,182 | 5,790 | 7,680 | 25,090 | 21,230 | 2,280 | 3,570 | 13,180 | 0,162 | 109,330 | -1 |
| 7 | 01.01.2020 1:40 | 92,00 | 1019,84 | -4,00 | 3,843 | 14,280 | 2,891 | 3,860 | 7,820 | 20,410 | 15,460 | 2,610 | 3,680 | 5,110 | 0,569 | 19,590 | 180 |
| 8 | 01.01.20 2:00 | 86,00 | 1019,86 | -4,00 | 2,164 | 19,570 | 2,029 | 2,940 | 7,550 | 12,730 | 15,760 | 3,360 | 3,440 | 8,480 | 1,094 | 96,390 | 225 |
| 9 | 01.01.2020 2:20 | 82,00 | 1019,21 | -4,00 | 0,572 | 25,310 | 2,524 | 10,600 | 4,100 | 22,320 | 32,560 | 2,520 | 8,080 | 9,310 | 0,179 | 8,630 | 90 |
| 10 | 01.01.20 2:40 | 84,00 | 1019,58 | -5,00 | 46,314 | 26,240 | 2,600 | 7,490 | 2,860 | 21,820 | 12,000 | 1,200 | 3,280 | 10,990 | 0,152 | 37,660 | 270 |
| 11 | 01.01.2020 3:00 | 85,00 | 1019,15 | -4,00 | 49,310 | 17,600 | 0,881 | 9,980 | 0,910 | 5,780 | 11,480 | 0,940 | 2,020 | 8,210 | 0,065 | 29,730 | -1 |
| 12 | 01.01.20 3:20 | 81,00 | 1019,02 | -4,00 | 34,108 | 26,530 | 2,207 | 7,820 | 7,930 | 17,130 | 10,930 | 0,490 | 3,310 | 3,690 | 0,124 | 54,620 | 225 |
| 13 | 01.01.2020 3:40 | 88,00 | 1019,39 | -4,00 | 48,694 | 29,980 | 1,450 | 3,600 | 6,040 | 11,880 | 8,380 | 3,470 | 0,700 | 7,600 | 1,341 | 18,590 | 1 |
| 14 | 01.01.20 4:00 | 86,00 | 1019,17 | -5,00 | 32,479 | 25,680 | 0,179 | 7,120 | 1,490 | 7,770 | 2,000 | 5,390 | 2,420 | 4,220 | 0,141 | 59,370 | 90 |
| 15 | 01.01.20 4:20 | 93,00 | 1019,90 | -4,00 | 47,154 | 20,920 | 1,385 | 2,630 | 5,910 | 11,630 | 2,580 | 2,610 | 1,760 | 34,600 | 0,146 | 42,880 | 45 |
| 16 | 01.01.2020 4:40 | 92,00 | 1019,48 | -4,00 | 2,219 | 15,890 | 2,577 | 6,970 | 1,650 | 25,220 | 4,050 | 1,400 | 1,960 | 8,300 | 0,532 | 92,340 | 225 |
| 17 | 01.01.20 5:00 | 94,00 | 1019,72 | -5,00 | 43,209 | 19,300 | 2,273 | 3,750 | 7,470 | 15,600 | 12,070 | 2,590 | 1,080 | 10,490 | 0,337 | 28,650 | 225 |
| 18 | 01.01.2020 5:20 | 84,00 | 1019,18 | -4,00 | 49,601 | 16,810 | 2,524 | 8,250 | 1,160 | 41,050 | 14,560 | 0,410 | 2,570 | 7,370 | 0,081 | 95,440 | 135 |

Рис. 3. Таблиця усереднення за 8 годин

Для створення таблиці усереднення за день використано функції, аналогічні до функцій усереднення за годину та за вісім годин:

=INDEX(Таблиця1[Дата та час];IF(ROW(Таблиця1[@[Дата та час]])>2;(ROW(Таблиця1[@[Дата та час]])-2)*72+2;ROW(Таблиця1[@[Дата та час]]))-1;0);

=AVERAGE(OFFSET(Таблиця1[[#Заголовки];[Вологість,
% RH]];IF(ROW([@[Дата та час]])>2;(ROW([@[Дата та час]])-2)*72+2;1);0;IF(ROW([@[Дата та час]])>2;72;73);1)).

Відрізняються функції лише коефіцієнтами кроку та розміру ковзкого вікна, що становить 72, і розміру першого вікна, що становить 73.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Дата та час | Вологість, % RH | Тиск, гПа | Температура, °C | Діоксид сірки, мкг/куб. м | Діоксид азоту та оксиди азоту, мкг/куб. м | Бензол, мкг/куб. м | Оксид вуглецю, мг/куб. м | Свинець, мкг/куб. м | Тверді частки 10, мкг/куб. м | Тверді частки 2,5, мкг/куб. м | Арсен, нг/куб. м | Кадмій, нг/куб. м | Нікель, нг/куб. м | Бенз(а)пірен, нг/куб. м | Озон, мкг/куб. | Напрямок вітру |
| 2 | 01.01.2020 | 87,41 | 1019,52 | -4,49 | 59,40 | 20,81 | 1,80 | 6,59 | 5,73 | 17,69 | 12,34 | 1,92 | 2,48 | 9,00 | 0,28 | 68,41 | 135,00 |
| 3 | 02.01.2020 | 86,85 | 1019,39 | -4,50 | 56,17 | 21,95 | 1,81 | 6,25 | 4,53 | 18,05 | 10,83 | 2,13 | 2,25 | 8,45 | 0,30 | 60,18 | 270,00 |
| 4 | 03.01.2020 | 87,24 | 1019,54 | -4,49 | 67,03 | 23,63 | 1,65 | 6,40 | 4,13 | 18,31 | 10,78 | 1,71 | 2,05 | 13,02 | 0,31 | 66,89 | 90,00 |
| 5 | 04.01.2020 | 88,00 | 1019,50 | -4,56 | 53,87 | 21,68 | 1,54 | 6,14 | 4,85 | 17,98 | 10,98 | 2,37 | 2,32 | 9,57 | 0,38 | 66,03 | -1,00 |
| 6 | 05.01.2020 | 87,04 | 1019,50 | -4,43 | 80,19 | 23,86 | 2,01 | 6,05 | 5,68 | 16,23 | 9,85 | 2,08 | 2,09 | 9,01 | 0,36 | 69,44 | 1,00 |
| 7 | 06.01.2020 | 88,06 | 1019,49 | -4,40 | 56,74 | 23,12 | 1,67 | 7,62 | 5,12 | 17,40 | 9,91 | 2,02 | 2,12 | 8,67 | 0,33 | 57,76 | 135,00 |
| 8 | 07.01.2020 | 87,32 | 1019,54 | -4,51 | 68,65 | 20,68 | 1,64 | 6,80 | 4,75 | 18,22 | 11,29 | 2,17 | 1,92 | 8,94 | 0,38 | 68,59 | 135,00 |
| 9 | 08.01.2020 | 86,75 | 1019,51 | -4,51 | 85,87 | 20,83 | 1,86 | 6,50 | 3,90 | 18,04 | 10,61 | 2,02 | 2,37 | 11,09 | 0,35 | 62,27 | 315,00 |
| 10 | 09.01.2020 | 87,36 | 1019,45 | -4,46 | 95,80 | 19,80 | 1,51 | 6,14 | 5,41 | 17,45 | 10,65 | 1,83 | 2,56 | 8,73 | 0,38 | 68,28 | 45,00 |
| 11 | 10.01.2020 | 86,64 | 1019,52 | -4,54 | 67,44 | 22,16 | 2,27 | 6,46 | 4,50 | 18,82 | 11,15 | 1,94 | 1,87 | 9,00 | 0,35 | 68,49 | 315,00 |
| 12 | 11.01.2020 | 87,42 | 1019,51 | -4,47 | 53,39 | 22,40 | 1,60 | 6,35 | 4,51 | 16,92 | 10,06 | 1,94 | 1,82 | 8,31 | 0,40 | 67,81 | 315,00 |
| 13 | 12.01.2020 | 87,96 | 1019,55 | -4,56 | 48,83 | 22,04 | 1,83 | 6,96 | 4,82 | 18,00 | 10,94 | 1,85 | 2,38 | 7,31 | 0,32 | 54,95 | 1,00 |
| 14 | 13.01.2020 | 87,07 | 1019,53 | -4,49 | 69,38 | 22,21 | 2,01 | 6,79 | 4,18 | 18,62 | 10,34 | 1,96 | 2,25 | 11,10 | 0,38 | 61,66 | 1,00 |
| 15 | 14.01.2020 | 88,49 | 1019,50 | -4,54 | 82,34 | 22,03 | 1,65 | 6,86 | 4,99 | 19,32 | 9,85 | 2,17 | 2,10 | 8,61 | 0,34 | 68,90 | 270,00 |
| 16 | 15.01.2020 | 88,10 | 1019,58 | -4,53 | 51,29 | 22,48 | 1,80 | 6,24 | 5,34 | 17,44 | 10,39 | 2,29 | 2,13 | 9,32 | 0,40 | 65,34 | 1,00 |
| 17 | 16.01.2020 | 87,44 | 1019,50 | -4,58 | 63,40 | 19,96 | 1,69 | 6,52 | 4,09 | 17,46 | 10,72 | 1,99 | 1,90 | 11,24 | 0,34 | 77,59 | 45,00 |

Рис. 4. Таблиця усереднення за добу

Визначення максимальних середньодобових восьмигодинних значень здійснюється подібно до визначення середніх значень за годину, але коренева функція, що визначає остаточне значення, відрізняється.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Дата та час | Вологість, % RH | Тиск, гПа | Температура, °C | Діоксид сірки, мкг/куб. м | Діоксид азоту та оксиди азоту, мкг/куб. м | Бензол, мкг/куб. м | Оксид вуглецю, мг/куб. м | Свинець, мкг/куб. м | Тверді частки 10, мкг/куб. м | Тверді частки 2,5, мкг/куб. м | Арсен, нг/куб. м | Кадмій, нг/куб. м | Нікель, нг/куб. м | Бенз(а)пірен, нг/куб. м | Озон, мкг/куб. | Напрямок вітру |
| 2 | 01.01.2020 | 87,75 | 1019,53 | -4,32 | 76,94 | 22,53 | 1,86 | 7,42 | 6,02 | 18,71 | 13,50 | 2,04 | 2,89 | 10,00 | 0,29 | 74,30 | #Н/Д |
| 3 | 02.01.2020 | 87,83 | 1019,44 | -4,42 | 65,15 | 22,81 | 1,84 | 6,53 | 5,12 | 18,71 | 11,44 | 2,27 | 2,37 | 9,87 | 0,34 | 68,37 | 270 |
| 4 | 03.01.2020 | 87,38 | 1019,61 | -4,38 | 75,60 | 27,54 | 1,84 | 6,91 | 4,55 | 18,65 | 12,05 | 1,92 | 2,32 | 14,26 | 0,38 | 73,62 | 1 |
| 5 | 04.01.2020 | 88,42 | 1019,54 | -4,42 | 94,22 | 22,07 | 1,99 | 6,70 | 5,08 | 18,91 | 11,79 | 2,75 | 2,59 | 11,39 | 0,41 | 70,65 | #Н/Д |
| 6 | 05.01.2020 | 88,00 | 1019,55 | -4,38 | 98,83 | 25,50 | 2,46 | 6,50 | 6,71 | 19,99 | 11,92 | 2,30 | 2,41 | 12,42 | 0,41 | 81,10 | #Н/Д |
| 7 | 06.01.2020 | 88,71 | 1019,50 | -4,33 | 63,87 | 25,00 | 2,42 | 7,92 | 5,80 | 19,13 | 10,21 | 2,11 | 2,54 | 9,91 | 0,40 | 59,62 | #Н/Д |
| 8 | 07.01.2020 | 88,21 | 1019,59 | -4,42 | 93,82 | 21,48 | 1,96 | 7,26 | 5,33 | 20,20 | 13,58 | 2,43 | 1,99 | 9,82 | 0,49 | 79,93 | #Н/Д |
| 9 | 08.01.2020 | 88,08 | 1019,53 | -4,42 | 106,19 | 21,05 | 2,11 | 6,74 | 4,02 | 19,38 | 11,48 | 2,33 | 2,63 | 13,62 | 0,42 | 66,33 | #Н/Д |
| 10 | 09.01.2020 | 88,04 | 1019,48 | -4,33 | 119,71 | 21,84 | 1,68 | 6,46 | 7,10 | 19,14 | 11,41 | 1,88 | 3,09 | 10,06 | 0,41 | 86,08 | #Н/Д |
| 11 | 10.01.2020 | 87,42 | 1019,60 | -4,46 | 90,42 | 23,98 | 2,42 | 6,47 | 5,18 | 19,28 | 12,59 | 2,07 | 2,03 | 10,59 | 0,36 | 73,94 | 315 |
| 12 | 11.01.2020 | 88,75 | 1019,57 | -4,42 | 87,21 | 22,79 | 1,70 | 6,69 | 4,90 | 18,74 | 10,30 | 2,06 | 2,13 | 9,28 | 0,47 | 71,78 | #Н/Д |
| 13 | 12.01.2020 | 88,67 | 1019,66 | -4,50 | 56,34 | 24,00 | 2,08 | 7,29 | 6,06 | 18,93 | 13,63 | 2,18 | 2,58 | 9,46 | 0,38 | 59,13 | #Н/Д |
| 14 | 13.01.2020 | 88,08 | 1019,60 | -4,46 | 83,21 | 24,59 | 2,15 | 7,46 | 4,39 | 20,18 | 11,49 | 2,53 | 2,49 | 12,78 | 0,47 | 63,90 | #Н/Д |
| 15 | 14.01.2020 | 88,96 | 1019,66 | -4,46 | 100,59 | 23,80 | 1,97 | 7,33 | 5,56 | 22,04 | 10,82 | 2,52 | 2,41 | 9,63 | 0,44 | 82,67 | 270 |

Рис. 5. Максимальні середньодобові восьмигодинні значення

Стовпець «Дата та час» визначається тією ж функцією, що і середньодобове значення:

=INDEX(Таблиця1[Дата та час];IF(ROW(Таблиця1[@[Дата та час]])>2;(ROW(Таблиця1[@[Дата та час]])-2)*72+2;ROW(Таблиця1[@[Дата та час]]))-1;0).

А в усіх інших стовпцях, де вимагається знаходження максимального значення за 8 годин, застосовано таку функцію:

=MAX(OFFSET(Таблиця3[[#Заголовки];[Вологість,

% RH]];(ROW([@[Дата та час]])-2)*3+1;0;3;1)).

Функція MAX() знаходить максимальне значення ковзкого діапазону з 3 комірок. Оскільки функція звертається до обробленої таблиці за 8 годин, то вже немає потреби робити окрему умову для першого рядка.

Усереднення за місяць є більш складною задачею, оскільки в кожному місці різна кількість днів, а в січні кількість днів відрізняється у високосні роки.

Для створення ковзкої області виділення до вихідного переліку стовпців таблиці додано допоміжні стовпці: «Допоміжний стовпець визначення дати», «Днів у місяці», «Усього днів пройшло з початку спостережень».

Виходячи з початкового припущення, що дані розпочинаються 01.01 об 00:00 будь-якого року, за допомогою функції, що записана в першу комірку стовпця «Днів у місяці», визначається кількість днів у першому місяці з таблиці з денним усередненням:

=DAY(DATE(YEAR(Таблиця2[@[Дата та час]]);MONTH(Таблиця2[@[Дата та час]])+1;1)-1).

Далі значення з першої комірки стовпця «Днів у місяці» записується в першу комірку «Усього днів пройшло з початку спостережень» за допомогою функції:

=$S2.

Потім визначається останній день місяця, на момент якого відбувається усереднення даних за допомогою функції:

=INDEX(Таблиця2[[#Усе];[Дата та час]];$T2+1;0).

Функція, що використовується для визначення останнього дня місяця, використовується в подальших комірках таблиці.

Для визначення кількості «Днів у місяці» з другого рядка і до кінця стовпця використовується функція, яка вказує на попередній рядок зі стовпця «Усього днів пройшло з початку спостережень»:

=DAY(DATE(YEAR(INDEX(Таблиця2[[#Усе];[Дата та час]];$T2+2;0));MONTH(INDEX

(Таблиця2[[#Усе];[Дата та час]];$T2+1;0))+2;1)-1).

Функція дає змогу зсунути виділення комірки дати на кількість рядків, що визначена в першому рядку стовпця «Усього днів пройшло з початку спостережень». Стовпець «Усього днів пройшло з початку спостережень» фактично є лічильником днів, який дає змогу зсувати ковзке вікно виділення для подальших розрахунків, оскільки вказує на рядок завершення попереднього виділення.

Для визначення значення стовпця «Усього днів пройшло з початку спостережень» з другого рядка використовується функція:

=$T2+$S3.

Зазначена функція додає значення попереднього рядка цього ж стовпця зі значенням поточного рядка стовпця «Днів у місяці».

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Дата та час | Вологість, % RH | Тиск, гПа | Температура, °C | Діоксид сірки, мкг/куб. м | Діоксид азоту та оксиди азоту, мкг/куб. м | Бензол, мкг/куб. м | Оксид вуглецю, мг/куб. м | Свинець, мкг/куб. м | Тверді частки 10, мкг/куб. м | Тверді частки 2,5, мкг/куб. м | Арсен, нг/куб. м | Кадмій, нг/куб. м | Нікель, нг/куб. м | Бенз(а)пірен, нг/куб. м | Озон, мкг/куб. | Напрямок вітру | Допоміжний стовпець визначення дати | Днів у місяці | Усього днів пройшло з початку спостережень |
| 2 | 31.01.2020 | 87,57 | 1019,50 | -4,49 | 63,37 | 22,00 | 1,82 | 6,45 | 4,68 | 18,23 | 10,49 | 2,04 | 2,05 | 9,44 | 0,34 | 65,56 | 180 | 31.01.2020 | 31,00 | 31,00 |
| 3 | 29.02.2020 | 87,53 | 1019,49 | -3,50 | 64,96 | 22,38 | 1,82 | 6,60 | 4,61 | 17,94 | 10,22 | 2,06 | 2,06 | 8,91 | 0,34 | 66,69 | 1 | 29.02.2020 | 29,00 | 60,00 |
| 4 | 31.03.2020 | 65,03 | 1019,50 | 1,50 | 67,72 | 21,73 | 1,78 | 6,57 | 4,59 | 17,83 | 10,57 | 2,05 | 2,13 | 8,85 | 0,34 | 65,84 | 90 | 31.03.2020 | 31,00 | 91,00 |
| 5 | 30.04.2020 | 65,07 | 1019,49 | 9,00 | 66,23 | 21,94 | 1,83 | 6,57 | 4,62 | 17,93 | 10,55 | 2,09 | 2,11 | 9,32 | 0,34 | 66,26 | -1 | 30.04.2020 | 30,00 | 121,00 |
| 6 | 31.05.2020 | 65,05 | 1019,51 | 15,00 | 58,59 | 21,75 | 1,82 | 6,52 | 4,65 | 17,97 | 10,42 | 2,03 | 2,14 | 8,94 | 0,34 | 65,92 | 270 | 31.05.2020 | 31,00 | 152,00 |
| 7 | 30.06.2020 | 47,98 | 1019,50 | 18,50 | 63,63 | 21,79 | 1,86 | 6,59 | 4,70 | 17,80 | 10,42 | 2,02 | 2,11 | 8,69 | 0,33 | 67,05 | 225 | 30.06.2020 | 30,00 | 182,00 |
| 8 | 31.07.2020 | 48,06 | 1019,51 | 19,50 | 68,19 | 22,14 | 1,83 | 6,53 | 4,63 | 17,75 | 10,33 | 2,04 | 2,11 | 9,52 | 0,34 | 65,25 | 180 | 31.07.2020 | 31,00 | 213,00 |
| 9 | 31.08.2020 | 48,01 | 1019,51 | 18,50 | 66,33 | 22,01 | 1,77 | 6,48 | 4,74 | 17,98 | 10,49 | 2,06 | 2,12 | 9,17 | 0,34 | 65,74 | 270 | 31.08.2020 | 31,00 | 244,00 |
| 10 | 30.09.2020 | 65,07 | 1019,49 | 13,51 | 65,76 | 21,94 | 1,86 | 6,66 | 4,60 | 18,11 | 10,29 | 2,07 | 2,13 | 9,08 | 0,34 | 66,39 | 1 | 30.09.2020 | 30,00 | 274,00 |
| 11 | 31.10.2020 | 64,94 | 1019,50 | 7,50 | 61,08 | 21,89 | 1,84 | 6,61 | 4,65 | 17,96 | 10,37 | 2,07 | 2,06 | 9,01 | 0,33 | 64,51 | 270 | 31.10.2020 | 31,00 | 305,00 |
| 12 | 30.11.2020 | 64,99 | 1019,51 | 1,47 | 68,56 | 22,08 | 1,79 | 6,62 | 4,62 | 18,00 | 10,49 | 2,08 | 2,12 | 9,48 | 0,35 | 67,36 | 1 | 30.11.2020 | 30,00 | 335,00 |

Рис. 6. Середні місячні значення

Потім для заповнення таблиці стовпець «Дати та час» копіюється з «Допоміжний стовпець визначення дати» за допомогою функції:

=$R2.

Значення усереднення та моди в інших стовпцях таблиці визначається майже аналогічно, як і в попередніх таблицях, з однією відмінністю: розмір виділення ковзкої області визначається з комірки стовпця «Днів у місяці», а значення початку виділення отримується з комірки попереднього рядка стовпця «Усього днів пройшло з початку спостережень». Типова функція має такий вигляд:

=AVERAGE(OFFSET(Таблиця2[[#Заголовки];[Вологість,

% RH]];IF(ROW([@[Дата та час]])>2;$T1+1;1);0;$S2;1)).

Таблиця усереднення за рік визначається за таблицею усереднення за місяць, оскільки місяців у кожному році дванадцять, то тут немає зайвого рядка даних, який потрібно враховувати під час роботи з вихідною таблицею, тому ковзка область виділення зсувається на 12 рядків середньомісячної таблиці кожен рядок поточної таблиці.

Для визначення дати останнього дня року використовується функція:

=INDEX(Таблиця6[Дата та час];(ROW([@[Напрямок вітру]])-1)*12;0).

Для розрахунку середнього значення використовується така типова функція:

=AVERAGE(OFFSET(Таблиця6[[#Заголовки];[Вологість,

% RH]];(ROW([@[Дата та час]])-2)*12+1;0;12;1)).

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Дата та час | Вологість, % RH | Тиск, гПа | Температура, °C | Діоксид сірки, мкг/куб. м | Діоксид азоту та оксиди азоту, мкг/куб. м | Бензол, мкг/куб. м | Оксид вуглецю, мг/куб. м | Свинець, мкг/куб. м | Тверді частки 10, мкг/куб. м | Тверді частки 2,5, мкг/куб. м | Арсен, нг/куб. м | Кадмій, нг/куб. м | Нікель, нг/куб. м | Бенз(а)пірен, нг/куб. м | Озон, мкг/куб. | Напрямок вітру | Допоміжний стовпець визначення дати | Днів у місяці | Усього днів пройшло з початку спостережень |
| 2 | 31.01.2020 | 87,57 | 1019,50 | -4,49 | 63,37 | 22,00 | 1,82 | 6,45 | 4,68 | 18,23 | 10,49 | 2,04 | 2,05 | 9,44 | 0,34 | 65,56 | 180 | 31.01.2020 | 31,00 | 31,00 |
| 3 | 29.02.2020 | 87,53 | 1019,49 | -3,50 | 64,96 | 22,38 | 1,82 | 6,60 | 4,61 | 17,94 | 10,22 | 2,06 | 2,06 | 8,91 | 0,34 | 66,69 | 1 | 29.02.2020 | 29,00 | 60,00 |
| 4 | 31.03.2020 | 65,03 | 1019,50 | 1,50 | 67,72 | 21,73 | 1,78 | 6,57 | 4,59 | 17,83 | 10,57 | 2,05 | 2,13 | 8,85 | 0,34 | 65,84 | 90 | 31.03.2020 | 31,00 | 91,00 |
| 5 | 30.04.2020 | 65,07 | 1019,49 | 9,00 | 66,23 | 21,94 | 1,83 | 6,57 | 4,62 | 17,93 | 10,55 | 2,09 | 2,11 | 9,32 | 0,34 | 66,26 | -1 | 30.04.2020 | 30,00 | 121,00 |
| 6 | 31.05.2020 | 65,05 | 1019,51 | 15,00 | 58,59 | 21,75 | 1,82 | 6,52 | 4,65 | 17,97 | 10,42 | 2,03 | 2,14 | 8,94 | 0,34 | 65,92 | 270 | 31.05.2020 | 31,00 | 152,00 |
| 7 | 30.06.2020 | 47,98 | 1019,50 | 18,50 | 63,63 | 21,79 | 1,86 | 6,59 | 4,70 | 17,80 | 10,42 | 2,02 | 2,11 | 8,69 | 0,33 | 67,05 | 225 | 30.06.2020 | 30,00 | 182,00 |
| 8 | 31.07.2020 | 48,06 | 1019,51 | 19,50 | 68,19 | 22,14 | 1,83 | 6,53 | 4,63 | 17,75 | 10,33 | 2,04 | 2,11 | 9,52 | 0,34 | 65,25 | 180 | 31.07.2020 | 31,00 | 213,00 |
| 9 | 31.08.2020 | 48,01 | 1019,51 | 18,50 | 66,33 | 22,01 | 1,77 | 6,48 | 4,74 | 17,98 | 10,49 | 2,06 | 2,12 | 9,17 | 0,34 | 65,74 | 270 | 31.08.2020 | 31,00 | 244,00 |
| 10 | 30.09.2020 | 65,07 | 1019,49 | 13,51 | 65,76 | 21,94 | 1,86 | 6,66 | 4,60 | 18,11 | 10,29 | 2,07 | 2,13 | 9,08 | 0,34 | 66,39 | 1 | 30.09.2020 | 30,00 | 274,00 |
| 11 | 31.10.2020 | 64,94 | 1019,50 | 7,50 | 61,08 | 21,89 | 1,84 | 6,61 | 4,65 | 17,96 | 10,37 | 2,07 | 2,06 | 9,01 | 0,33 | 64,51 | 270 | 31.10.2020 | 31,00 | 305,00 |
| 12 | 30.11.2020 | 64,99 | 1019,51 | 1,47 | 68,56 | 22,08 | 1,79 | 6,62 | 4,62 | 18,00 | 10,49 | 2,08 | 2,12 | 9,48 | 0,35 | 67,36 | 1 | 30.11.2020 | 30,00 | 335,00 |

Рис. 7. Середньорічні значення

Частини звіту розподілені відповідно до пунктів Додатку 2 до Порядку здійснення державного моніторингу в галузі охорони атмосферного повітря.

Перша частина звіту про підрахунок кількості перевищень встановлених нормативів формується за пунктом 2 Додатку 2 до Порядку під назвою «Верхній та нижній пороги оцінювання».

Аркуш звіту умовно поділений на 3 стовпці: «Діапазон оцінювання», «Забруднювач», «Нормативи».

У звіті передбачено 4 типи задання діапазону оцінювання за датами: за дні, за години, за роки, за 8 годин. Для задання необхідного діапазону оцінювання потрібно вписати у комірках стовпця «Діапазон оцінювання» дату й час у форматі та з обмеженнями:

1) за годинного аналізу:

– початкова дата та час для поточного датасету: 01.01.2020 1:00:00;

– кінцева дата та час: в межах поточного датасету з кроком у годину (2:00:00, 3:00:00);
– початкова дата для будь-якого датасету виходячи з вихідного припущення 01.01 (будь-якого року від 1900 до 9999) 1:00:00;
2) за восьмигодинного аналізу:
– початкова дата та час для поточного датасету: 01.01.2020 8:00:00;
– кінцева дата та час: у межах поточного датасету з кроком у вісім годин (0:00:00, 16:00:00);
– початкова дата для будь-якого датасету виходячи з вихідного припущення 01.01 (будь-якого року від 1900 до 9999) 8:00:00;
3) у разі аналізу за дні:
– початкова дата для поточного датасету: 01.01.2020;
– кінцева дата: у межах поточного датасету з кроком у день (02.03.2020, 02.03.2023);
– початкова дата для будь-якого датасету виходячи з вихідного припущення 01.01 (будь-якого року від 1900 до 9999);
4) у разі аналізу за роки:
– початкова дата для поточного датасету: 31.12.2020;
– кінцева дата: у межах поточного датасету з кроком у рік (31.12.2021, 31.12.2023);
– початкова дата для будь-якого датасету виходячи з вихідного припущення 31.12 (будь-якого року від 1900 до 9999).

Для можливості зміни контрольних значень нормативів, за якими відбувається підрахунок перевищень у таблиці, а також для підсвітки підрахованої кількості перевищень, що більше за допустиму, використовується умовне кольорове форматування, яке реалізується функцією:

=$E$4>=$J$4,

до комірки =$E$4. Аналогічне форматування застосоване до всіх комірок, які передбачають порівняння кількості допустимих перевищень із виявленими.

Підрахунок кількості перевищень здійснюється типовою функцією:

=COUNTIF(OFFSET(Таблиця2[Діоксид сірки, мкг/куб. м];XMATCH(A3;Таблиця2[Дата та час])+1-2;0;XMATCH(B3;Таблиця2[Дата та час])+1-XMATCH(A3;Таблиця2[Дата та час]);1);">="&$I$3).

Особливістю функції у застосуванні до інших комірок є заміна адрес, що вказують на нормативи, діапазон цільових дат, а також вказання на таблицю усереднення, з якої здійснюється підрахунок перевищень. Подана функція підраховує кількість перевищень у стовпці «Діоксид сірки, мкг/куб. м» таблиці усереднення за день.

Функція для розрахунку перевищень дещо подібна до функцій усереднення та базується на ковзкій області виділення. Для задання відступу від початку цільової таблиці усереднення використовується функція:

XMATCH(A3;Таблиця2[Дата та час])+1-2,

що повертає кількість рядків від початку таблиці до початку діапазону аналізу. Висота ковзкого виділення визначається функцією:

XMATCH(B3;Таблиця2[Дата та час])+1-XMATCH(A3;Таблиця2[Дата та час]),

яка повертає різницю рядків між кінцем діапазону оцінювання та його початком.

Функція COUNTIF() дає змогу підрахувати кількість рядків, що перевищують норматив, указаний у комірці $I$3.

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | **Діапазон оцінювання** | | | | **Забруднювач** | | | | | | **Нормативи** | |
| 2 | **З** | **По** | | **Діоксид сірки** | Охорона здоров'я, кількість перевищень | Захист рослинності, кількість перевищень | | | Охорона здоров'я, мкг/куб. м | Допустимо перевищень | Захист рослинності, мкг/куб. м | Допустимо перевищень |
| 3 | 01.01.2020 | 31.12.2020 | | Верхній поріг оцінювання | 80 | 366 | | | 75 | 3 | 12 | – |
| 4 | | | | Нижній поріг оцінювання | 293 | 366 | | | 50 | 3 | 8 | – |
| 5 | | | | | | | | | | | | |
| 6 | **З** | **По** | | **Діоксид азоту та оксиди азоту** | Щогодинна гранична величина для захисту здоров'я людини | | | | Щогодинна гранична величина для захисту здоров'я людини, мкг/куб. м | Допустимо перевищень | | |
| 7 | 01.01.20 1:00 | 01.05.20 1:00 | | Верхній поріг оцінювання | 0 | | | | 140 | 18 | | |
| 8 | | | | Нижній поріг оцінювання | 0 | | | | 100 | 18 | | |
| 9 | | | | | | | | | | | | |
| 10 | **З** | **По** | | **Діоксид азоту та оксиди азоту** | Щорічна гранична величина для захисту здоров'я людини | Щорічний критичний рівень для захисту рослинної та природної екосистем | | | Щорічна гранична величина для захисту здоров'я людини, мкг/куб. м | Допустимо перевищень | Щорічний критичний рівень для захисту рослинної та природної екосистем, мкг/куб. м | Допустимо перевищень |
| 11 | 31.12.2020 | 31.12.2023 | | Верхній поріг оцінювання | 0 | 0 | | | 32 | – | 24 | – |
| 12 | | | | Нижній поріг оцінювання | 0 | 4 | | | 26 | – | 19,5 | – |
| 13 | | | | | | | | | | | | |

Рис. 8. Перша частина звіту

Друга частина звіту про підрахунок кількості перевищень граничних величин встановлених нормативів формується за пунктом 3 Додатку 2 до Порядку під назвою «Граничні величини забруднювальних речовин», а третя частина – за пунктом 4 під назвою «Інші рівні забруднювальних речовин, за якими проводиться оцінка якості атмосферного повітря».

У частині простого підрахунку кількості перевищень відносно встановленого нормативу в таблиці за певний період усереднення створені функції працюють так само, як і в першій частині звіту.

Відрізнятися у функціях буде підрахунок перевищень, що мають спостерігатися протягом послідовних трьох годин. Для цього посилання у функції йде не на цільовий стовпець таблиці з годинним усередненням, а на допоміжний стовпець цієї таблиці за цим маркером. Приклад функції для підрахунку перевищень діоксиду сірки протягом трьох послідовних годин рівня 500 мкг/куб.:

=COUNTIF(OFFSET(Таблиця4[Допоміжний стовпець для розрахунку перевищень діоксиду сірки ВНП827ч4];XMATCH(A24;Таблиця4[Дата та час])+1-2;0;XMATCH(B24;Таблиця4[Дата та час])+1-XMATCH(A24;Таблиця4[Дата та час]);1);"="&1).

У цій функції немає посилання безпосередньо на рівень, встановлений у нормативі, адже він використовується у формулі для визначення значення допоміжного стовпця, а порівняння здійснюється відносно 1 або 0.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 23 | З | По | | **Діоксид сірки** | Три послідовні години | | Одна година, мкг/куб. м | Допустимо перевищень |
| 24 | 01.01.20 1:00 | 01.05.20 1:00 | | Кількість перевищень | 0 | | 500 | – |
| 25 | | | | | | | | |
| 26 | З | По | | **Діоксид азоту** | Три послідовні години | | Одна година, мкг/куб. м | Допустимо перевищень |
| 27 | 01.01.20 1:00 | 01.05.20 1:00 | | Кількість перевищень | 0 | | 400 | – |
| 28 | | | | | | | | |
| 29 | З | По | | **Озон** | Три послідовні години | | Одна година, мкг/куб. м | Допустимо перевищень |
| 30 | 01.01.20 1:00 | 01.05.20 1:00 | | Кількість перевищень | 0 | | 240 | – |
| 31 | | | | | | | | |
| 32 | З | По | | **Озон-інформування** | Три послідовні години | | Одна година, мкг/куб. м | Допустимо перевищень |
| 33 | 01.01.20 1:00 | 01.05.20 1:00 | | Кількість перевищень | 0 | | 180 | – |

Рис. 9. Підрахунок маркерів за три послідовні години

**Висновки**

Під час проведення дослідження було розроблено метод підрахунку ключових екологічних маркерів для автоматичного формування звітності за постановою № 827 КМУ. Метод базується на основі використання ковзкої області виділення комірок і припущення, що датасет починається з 01.01 0:00:00. За допомогою методу можна підраховувати кількість перевищень екологічних маркерів за заданий діапазон часу в межах датасету з різним діапазоном усереднення й урахуванням перевищень нормативів протягом трьох послідовних годин.

Наукова новизна роботи полягає в тому, що вперше розроблено метод підрахунку ключових екологічних маркерів для автоматичного формування звітності за постановою № 827 КМУ, що базується на основі використання ковзкої області виділення комірок і припущення, що датасет починається з 01.01 0:00:00.

Практичним значенням проведеного дослідження є те, що оператор з НДЦ може замінити генеровані дані у вихідній 20-хвилинній таблиці Excel на архівні дані з конкретної станції, вибрати потрібний проміжок часу аналізу перевищень та на виході отримати підраховану кількість перевищень маркерами атмосферного повітря встановлених нормативів, що значно спростить процес аналізу даних.

У подальших дослідженнях планується створення інформаційної системи для збирання, обробки та збереження даних зі станцій EcoCity, що дасть змогу реалізувати систему нижчого рівня екологічного моніторингу, яку можна буде використовувати для виявлення й передбачення надзвичайних ситуацій воєнного характеру.

***Література***

1. Деякі питання здійснення державного моніторингу в галузі охорони атмосферного повітря. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/827-2019-п#Text (дата звернення: 23.10.2023).

2. DOAJ. Directory Of Open Access Journals. URL: https://doaj.org/ (дата звернення: 23.10.2023).

3. ScienceDirect. Search for peer-reviewed journal articles and book chapters (including open access content). URL: https://www.sciencedirect.com/ (дата звернення: 23.10.2023).

4. Crossref. Search the metadata of journal articles, books, standards, datasets & more. URL: https://search.crossref.org/ (дата звернення: 23.10.2023).

5. Google Scholar. URL: https://scholar.google.com/ (дата звернення: 23.10.2023).

6. DuckDuckGo. URL: https://duckduckgo.com/ (дата звернення: 23.10.2023).
7. Коростельов А. С., Гученко М. І., Перекрест А. Л., Нікітіна А. В., Вадурін К. О. Модель корпоративної мережі базованої на технологіях інтернету речей підприємства з екологічних досліджень. Системи управління, навігації та зв'язку. 2023. Т. 3. №. 73. С. 111–114. DOI: https://doi.org/10.26906/SUNZ.2023.3.111.

### *References*

1. *Deiaki pytannia zdiisnennia derzhavnoho monitorynhu v haluzi okhorony atmosfernoho povitria. Ofitsiinyi vebportal parlamentu Ukrainy [Some issues of state monitoring in the field of atmospheric air protection. Official web portal of the Parliament of Ukraine]*. Retrieved from: https://zakon.rada.gov.ua/laws/show/827-2019-п#Text [in Ukrainian].
2. *DOAJ. Directory Of Open Access Journals.* Retrieved from: https://doaj.org/ [in English].
3. *ScienceDirect. Search for peer-reviewed journal articles and book chapters (including open access content).* Retrieved from: https://www.sciencedirect.com/ [in English].
4. *Crossref. Search the metadata of journal articles, books, standards, datasets & more.* Retrieved from: https://search.crossref.org/ [in English].
5. *Google Scholar.* Retrieved from: https://scholar.google.com/ [in English].
6. *DuckDuckGo.* Retrieved from: https://duckduckgo.com/ [in English].
7. Korostelov, A., Guchenko, M., Perekrest, A., Nikitina, A., & Vadurin, K. (2023). *Model korporatyvnoi merezhi bazovanoi na tekhnolohiiakh internetu rechei pidpryiemstva z ekolohichnykh doslidzhen [A model of a corporate network based on iot technologies of an environmental research enterprise]*. Control, Navigation and Communication Systems. 3 (73), 111–-114. DOI: https://doi.org/10.26906/SUNZ.2023.3.111 [in Ukrainian].

**UDC 519.8**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-06**

# CALCULATION OF CHARACTERISTICS OF QUEUING SYSTEMS USING THE ERLANG METHOD AND CONSERVATION LAWS

***Hanna Dymova,*** *Candidate of Technical Sciences, Phd., Associate Professor,*
*Department of Management, Marketing and Information Technology,*
*Kherson State Agrarian and Economic University, Kherson/Kropyvnytskyi, Ukraine*
*https://orcid.org/0000-0002-5294-1756, anndymova@gmail.com*

***Abstract.*** *Any human activity (economic, entrepreneurial, commercial) is associated with the performance of many operations at the stages of movement of raw materials, goods, products from the supply sector to the sphere of production and consumption. Such operations are transportation, storage, processing, sales, etc. These types of activities are characterized by the massive receipt of products, goods, money, and clients at random times; their sequential servicing is carried out by corresponding operations, the execution time of which, as a rule, is also random. All this leads to inequalities in work, creates idle time, underload and overload in operations. Therefore, the tasks arise of analyzing existing options for performing a certain set of operations, identifying bottlenecks and reserves for developing and making management decisions to improve the operating efficiency of any organization. Such problems are successfully solved using queuing theory. The basis for making management decisions are parameters that characterize different aspects of the operation of both the system as a whole and its individual elements. To describe individual parts of the system, the distribution of random variables and their numerical properties are traditionally used. The main characteristics of the action and state of the QS are the average number of requests in the queue or system, the average waiting time for service and others, as well as the value of some probabilities (the probability of service denial, the probability that there are at least a certain number of requirements in the system, the probability that the system is free from maintenance, etc.).*

*The article describes some methods and techniques for studying Markov systems (the processes of which have no history), which turn out to be applicable to more general systems. A Markov queuing system with a restriction on the largest number of requirements in the system is considered. The probabilities of state transitions for closed and open-loop systems are described and calculated using Erlang methods and using the laws of conservation of system probabilities and the Laplace transform. Calculations of queue safety under stationary operating modes of queuing systems are demonstrated.*

***Key words:*** *queuing systems, Markov systems, Erlang method, closed-loop system, open-loop system, conservation laws.*

# РОЗРАХУНОК ХАРАКТЕРИСТИК СИСТЕМ МАСОВОГО ОБСЛУГОВУВАННЯ ІЗ ЗАСТОСУВАННЯМ МЕТОДУ ЕРЛАНГА І ЗАКОНІВ ЗБЕРЕЖЕННЯ

***Димова Г.О.,*** *канд. техн. наук, Phd., доцент кафедри менеджменту, маркетингу*
*та інформаційних технологій, Херсонський державний аграрно-економічний університет,*
*Херсон/Кропивницький, Україна*
*https://orcid.org/0000-0002-5294-1756, anndymova@gmail.com*

***Анотація.*** *Будь-яка діяльність людини (господарська, підприємницька, комерційна) пов'язана з виконанням безлічі операцій на етапах руху сировини, товарів, виробів зі сфери постачання до сфери виробництва та споживання. Такими операціями є транспортування, зберігання, обробка, реалізація тощо. Ці види діяльності характеризуються масовістю надходження виробів, товарів, грошей, клієнтів у випадкові моменти часу, їх послідовне обслуговування здійснюється відповідними операціями, час виконання яких здебільшого теж має випадковий характер. Усе це призводить до нерівностей у роботі, породжує простої, недовантаження та перевантаження в операціях. Тому виникають задачі аналізу існуючих варіантів виконання деякої сукупності операцій, виявлення вузьких місць і резервів для розробки та прийняття управлінських рішень щодо підвищення ефективності функціонування будь-якої організації. Такі задачі успішно розв'язуються за допомогою теорії масового обслуговування. Основою для прийняття управлінських рішень є параметри, які характеризують різні сторони дії як системи загалом , так і окремих її елементів. Для опису окремих елементів системи зазвичай використовують розподіл випадкових величин та їх числові характеристики. Основними характеристиками дії і стану СМО є середнє число вимог у черзі або в системі, середній час очікування обслуговування й інші, а також значення деяких імовірностей (імовірності відмови в обслуговуванні, ймовірності того, що в системі розміщено не менше за певне число вимог, імовірності того, що система вільна від обслуговування тощо).*

*У статті описані деякі методи і прийоми дослідження марковських систем (процеси яких не мають передісторії), що виявляються застосованими до більш загальних систем. Розглядається марковська система масового обслуговування з обмеженням максимальної кількості вимог у системі. Описані й розраховані ймовірності переходів станів для замкнених і розімкнених систем методами Ерланга та з використанням законів збереження ймовірностей систем і перетворення Лапласа. Продемонстровані обчислення збереження черги за стаціонарних режимів роботи систем масового обслуговування.*

***Ключові слова:*** *системи масового обслуговування, марковські системи, метод Ерланга, замкнена система, розімкнена система, закони збереження.*

**Introduction**

When researching operations, one often has to deal with the operation of peculiar systems called queuing systems (QS). Examples of such systems include: electronic queues, repair shops, ticket offices, queues for family doctors, shops, hairdressers, computer networks.

The QS may consist of a certain number of service units (devices), which are called service channels. Channels can be: communication lines, workplaces, cashiers, salespeople, elevators, cars, etc. QS can be single-channel or multi-channel.

The QS is designed to service the flow of applications (requirements) arriving at its input at random moments in time. Servicing of the request continues for a random time $T_s$, after which the channel is freed and ready to receive the next request. The random nature of the flow of applications and service times leads to the fact that a large number of applications can accumulate at the input of the QS (they either queue up or leave the QS unserved); in other periods, the QS will work with underload or be completely idle [1].

The QS operation process is a random process with discrete states and continuous time; the state of the QS changes abruptly at the moments of occurrence of events (the arrival of a new request, the end of service, or the moment when an application that is tired of waiting leaves the queue, applications with a limited waiting time).

The subject of the theory of queuing systems is the construction of mathematical models that connect the given operating conditions of the QS (the number of channels, their productivity, operating rules, the nature of the flow of requests) with the performance indicators of interest to the QS, which can be divided into two groups: indicators directly related to the stationary probability distribution $\{P_k\}$ number of applications in the system and time indicators.

The theory of queuing began with the work of the Danish scientist A.K. Erlang for calculating telephone networks. In the formation and development of this theory, a prominent role belongs to scientists A.A. Markov, A.Ya. Khinchin, A.N. Kolmogorov, B.V. Gnedenko, I.N. Kovalenko and many others [2, 3].

Despite the fact that methods have now been developed and classes of QS that are much more general than the class of Markov QS have been quite fully studied, a separate study of the latter is important. This is explained, on the one hand, by the fact that some methods and techniques for studying Markov systems turn out to be applicable (in a modified form) to more general systems and, on the other hand, by the simplicity and clarity of formulas expressing the characteristics of a system through its parameters. This allows us to better understand the qualitative behavior of service processes and evaluate the impact of changing system parameters on the characteristics of interest [2, 4].

**Research**

***Calculation of Markov systems.*** If the future behavior of a process depends on its past history, then by appropriately expanding the concept of "state," in other words, by increasing the number of components of the vector representing the process, it can be reduced to a Markovian one.

Distribution of intervals for the simplest incoming stream $A(\theta)=1-e^{-\lambda\theta}$, exponential distribution of service time $B(\theta)=1-e^{-\mu\tau}$.

***Erlang method.*** Let's consider a Markov queuing system with a limit on the maximum number of requests in the system by value $R$. The states of such a system can be characterized by the number of requirements of applications. Possible transitions between states are indicated in Figure 1, and the corresponding probabilities are in Table 1. There is no maintenance in state $E_0$ (system idle). Applications that find the system in the $E_R$ state are rejected [2, 3].

The probabilities of making two or more jumps are of the order of smallness, higher than $\Delta t$, and are not included in the table.

Table 1

Probabilities of transitions in a Markov system over an interval $\Delta t$

| State of the system | | Transition probability accurate to 0 ($\Delta$t) |
|---|---|---|
| **initial** | **final** | |
| $\mathrm{E}_0(t)$ | $\mathrm{E}_1(t+\Delta t)$ | $\lambda_0 \Delta t$ |
| $\mathrm{E}_0(t)$ | $\mathrm{E}_0(t+\Delta t)$ | $1-\lambda_0 \Delta t$ |
| $\mathrm{E}_k(t), k=\overline{1, R-1}$ | $\mathrm{E}_{k-1}(t+\Delta t)$<br>$\mathrm{E}_{k-1}(t+\Delta t)$<br>$\mathrm{E}_k(t+\Delta t)$ | $\mu_k \Delta t$<br>$\lambda_k \Delta t$<br>$1-(\lambda_k+\mu_k)\Delta t$ |
| $\mathrm{E}_R(t)$ | $\mathrm{E}_{R-1}(t+\Delta t)$<br>$\mathrm{E}_{R-1}(t+\Delta t)$ | $\mu_R \Delta t$<br>$1-\mu_R \Delta t$ |

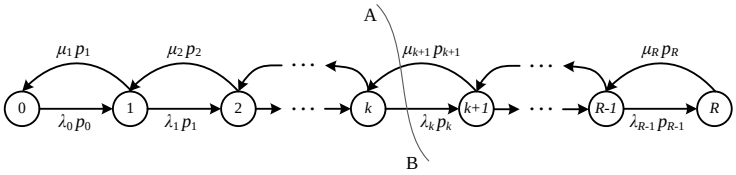


Fig. 1. Law of conservation of state probabilities for Markov systems

Combining the probabilities of events, according to the total probability theorem, we obtain a system of difference equations

$$\begin{cases} p_0(t+\Delta t)=p_0(t)(1-\lambda_0\Delta t)+\mu_1\Delta t p_1(t)+0(\Delta t) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ p_k(t+\Delta t)=p_k(t)[1-(\lambda_k+\mu_k)\Delta t]+\lambda_{k-1}\Delta t p_{k-1}(t)+ \\ \quad +\mu_{k+1}\Delta t p_{k+1}(t)+0(\Delta t), \qquad k=\overline{1, R-1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ p_R(t+\Delta t)=p_R(t)(1-\mu_R\Delta t)+\lambda_{R-1}\Delta t p_{R-1}(t)+0(\Delta t) \end{cases} \tag{1}$$

Let's regroup the terms of these equations and divide both sides of each of them by $\Delta t$:

$$\begin{cases} \dfrac{p_0(t+\Delta t)-p_0(t)}{\Delta t}=-\lambda_0 p_0(t)+\mu_1 p_1(t)+\dfrac{0(\Delta t)}{\Delta t} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \dfrac{p_k(t+\Delta t)-p_k(t)}{\Delta t}=-(\lambda_k+\mu_k)p_k(t)+\lambda_{k-1}p_{k-1}(t)+ \\ \quad +\mu_{k+1}p_{k+1}(t)+\dfrac{0(\Delta t)}{\Delta t}, \qquad k=\overline{1, R-1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \dfrac{p_R(t+\Delta t)-p_R(t)}{\Delta t}=-\mu_R p_R(t)+\lambda_{R-1}p_{R-1}(t)+\dfrac{0(\Delta t)}{\Delta t} \end{cases} \tag{2}$$

Letting $\Delta t$ tend to zero, we obtain a system of differential equations

$$\begin{cases} p_0'(t)=-\lambda_0 p_0(t)+\mu_1 p_1(t) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ p_k'(t)=-(\lambda_k+\mu_k)p_k(t)+\lambda_{k-1}p_{k-1}(t)+ \\ \quad +\mu_{k+1}p_{k+1}(t),\ k=\overline{1, R-1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ p_R'(t)=-\mu_R p_R(t)+\lambda_{R-1}p_{R-1}(t) \end{cases} \tag{3}$$

By specifying the initial conditions for system (3), for example, in the form $p_0(0)=1$ and $p_k(0)=0$ for $k=\overline{1,R}$, one can find a numerical solution to the corresponding Cauchy problem for an arbitrary value of *t*.

As $t\to\infty$, the time derivatives, if a stationary regime exists, vanish. As a result, we arrive at a system of linear algebraic equations for stationary probabilities

$$\begin{cases} -\lambda_0 p_0+\mu_1 p_1=0 \\ \lambda_0 p_0-(\lambda_1+\mu_1)p_1+\mu_2 p_2=0 \\ \lambda_1 p_1-(\lambda_2+\mu_2)p_2+\mu_3 p_3=0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \lambda_{k-1}p_{k-1}-(\lambda_k+\mu_k)p_k+\mu_{k+1}p_{k+1}=0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \lambda_{R-1}p_{R-1}-\mu_R p_R=0 \end{cases} \tag{4}$$

Adding the first *k* + 1 equations of this system, we are convinced that

$$-\lambda_k p_k+\lambda_{k+1}p_{k+1}=0 \tag{5}$$

whence it follows

$$p_{k+1}=\frac{\lambda_k}{\mu_{k+1}}p_k=\frac{\lambda_k}{\mu_{k+1}}\frac{\lambda_{k-1}}{\mu_k}p_{k-1}=\ldots=\frac{\lambda_k}{\mu_{k+1}}\frac{\lambda_{k-1}}{\mu_k}\ldots\frac{\lambda_2}{\mu_2}p_1. \tag{6}$$

From the first equation we get $p_0=\frac{\lambda_0}{\mu_0}p_0$. Thus, for a problem with transitions only to neighboring states (the "reproduction and death" scheme), all probabilities are expressed through $p_0$ according to

$$p_k=\alpha_k p_0,\ k=\overline{1,R},$$

where $\alpha_k=\prod_{i=0}^{k-1}\frac{\lambda_i}{\mu_{i+1}}$.

The probability $p_0$ is determined from the normalization condition (the sum of all probabilities must be equal to one). Consequently,

$$p_0=\left(1+\sum_{k=1}^{R}\alpha_k\right)^{-1} \tag{7}$$

From the considered scheme, various special cases can be obtained, for example, $\lambda_K=\lambda(R-K)K$ (closed-loop system), $\lambda_K=\lambda$ (open-loop system), as well as the case of an *n*-linear system, for which

$$\mu_k=\begin{cases} k\mu,k=\overline{1,n-1} \\ n\mu,\ k=\overline{1,R} \end{cases}.$$

For an open-loop system, $R=\infty$ corresponds to an unlimited queue. For $R=\infty$, the condition for the existence of a stationary solution is the convergence of the series $\sum_{K=1}^{\infty}\alpha_K$. For the series to converge, it is sufficient to require the existence of a constant $q\in(0,1)$, which exceeds any ratio $\frac{\lambda_K}{\mu_1}p_0$, starting from some number *k*. In this case, subsequent terms of the series are majorized by terms of a decreasing geometric progression with denominator *q*, the sum of which is finite.

The final Erlang formulas for the open-loop system $M/M/n$:

$$\begin{cases} p_0=\left[1+\sum_{i=1}^{n}\left(\frac{\lambda}{\mu}\right)^i\frac{1}{i!}+\left(\frac{\lambda}{\mu}\right)^n\frac{\lambda}{n!(n\mu-\lambda)}\right]^{-1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ p_k=p_0\left(\frac{\lambda}{\mu}\right)^k\frac{1}{k!},\ k=\overline{1,n} \\ p_0=p_n\left(\frac{\lambda}{n\mu}\right)^{k-n},\ k=n+1,n+2,\ldots \end{cases} \tag{8}$$

For systems with a finite number of states, especially closed-loop ones, $p_0=1$, and using formula (5) we calculate $\{p_k\}$ for specific dependencies $\mu(k)$ and $\lambda(k)$, simultaneously calculating the sum, and then normalize the results.

For the abbreviated designation of QS, D. Kendall proposed using a formula of the form $A/B/n/\mathrm{R}$, where *A* indicates the distribution of intervals between requests, *B* is the distribution of service time, *n* is the number of channels, R is the maximum number of requests in the system [3, 5].

***Calculation of the** M / M / 1 **system using the laws of conservation of state probabilities.*** Let us apply the laws of conservation of state probabilities to the calculation of the main indicators of the $\mathrm{M/M/1}$ system (single-channel QS with the simplest incoming flow and exponential distribution with the Markov property, with an unlimited queue).

First of all, we note that for this QS $\lambda_K = \lambda$ and $\mu_K = \mu$ and does not depend on *k*. Consequently, for all *K* we have $\mu p_{K+1} = \lambda p_k$, or $p_{K+1} = \frac{\lambda}{\mu} p_k$. By introducing the notation $x = \frac{\lambda}{\mu}$ – system load factor, we obtain formulas for stationary probabilities $p_k = (1-x)x^k, k = 0,1,\ldots$

The condition for the existence of a stationary regime is the inequality $\lambda < \mu$. Note that $\frac{\lambda}{\mu} = \bar{\tau}$ ($\bar{\tau}$ is the average service time). Consequently, $p_0 = 1 - \frac{\lambda}{\mu} = 1 - \lambda\bar{\tau}$, which confirms the law of conservation of requirements $P\left(1 - \frac{s}{\lambda}\right) = \upsilon(s)$.

Generating function of probabilities

$$P(z) = \sum_{k=0}^{\infty} (1-x)x^k z^k = \frac{1-x}{1-xz} \tag{9}$$

Based on the formula, the Laplace transform of the distribution of the application's residence time in the system

$$\upsilon(s) = P\left(1 - \frac{s}{\lambda}\right) = \frac{1-x}{1-x\left(1-\frac{s}{\lambda}\right)} = \frac{\mu-\lambda}{\mu-\lambda+s} \tag{10}$$

Consequently, the residence time of a request in the system is subject to the exponential law with the parameter $\lambda - \mu$. Average stay time

$$\upsilon_1 = \frac{1}{\mu-\lambda},$$

as $\lambda \to \mu$ increases without limit. This leads to the conclusion that it is inadmissible to choose an average service speed equal to the intensity of the flow of requests. Average number of applications in the system

$$L_1 = (1-x)\sum_{k=1}^{\infty} kx^k = x(1-x)\sum_{k=1}^{\infty} kx^{k-1} =$$

$$= x(1-x)\frac{d}{dx}\frac{1}{(1-x)} = \frac{x}{(1-x)} = \frac{\lambda}{\mu-\lambda}$$

Thus, the relation $L[1]=\lambda v_1$ *u* between $L_1$ and $v_1$ turned out to be correct, which is consistent with the queue conservation law.

**Conclusion**

The paper considers the possibility of using the Erlang method to construct models of queuing systems with a limit on the number of requests associated with the system.

Calculations of queue conservation under stationary operating modes of the M / M / 1 type QS using conservation laws and Laplace transforms, generating distribution functions are shown.

***Література***

1. Saaty T.L. Elements of Queueing Theory, with Applications. Published by Dover Publications. 1983, 423 p. ISBN 10: 0486645533 / ISBN 13: 9780486645537.

2. Математичні основи теорії телекомунікаційних систем. / В. В. Поповський, С. О. Сабуров, В. Ф. Олійник, Ю. І. Лосєв, Д. В. Агеєв [та ін.] ; за заг. ред. В. В. Поповського. Харків : ТОВ «Компанія СМІТ», 2006. 564 с. ISBN 966-8530-60-8.

3. Ларіонов Ю. І., Марченко Л. С., Хаджмуратов М. А. Дослідження операцій : навч. посіб. для вузів. Частина 2. Харків : Інжек, 2005. 288 с. ISBN 966-8515-91-9.

4. Димова Г. О., Ларченко О. В. Моделі і методи інтелектуального аналізу даних : навчальний посібник. Херсон : Книжкове видавництво ФОП Вишемирський В. С., 2021. 142 с. ISBN 978-617-7941-60-5.

5. Moor J.H., Weatherford L.R. Decision Modeling With Microsoft Excel. Upper Sddle River, New Jersey 07458: Published by Prentice Hall, Inc. 2001, 693 p. ISBN 0-13-017789-X.

### *References*

1. Saaty, Thomas L. (1983). Elements of Queueing Theory, with Applications. Published by Dover Publications. ISBN 10: 0486645533 / ISBN 13: 9780486645537.

2. Popovs'kyy, V.V.; Saburova, S.O.; Oliynyk, V.F.; Losyev, YU.I.; Aheyev, D.V. & ta in. (2006). Matematychni osnovy teoriyi telekomunikatsiynykh system [Mathematical foundations of the theory of telecommunication systems]. Kharkiv: Kompaniya SMIT. ISBN 966-8530-60-8 [in Ukrainian].

3. Larionov, Yu.I.; Marchenko, L.S. & Hazhmuradov, M.A. (2005). Doslidzhennya operatsiy. Chastyna II: Navchal'nyy posibnyk [Operational Research. Part II: Study Guide]. Kharkiv: ENGEC PH. ISBN 966-8515-91-9 [in Ukrainian].

4. Dymova, H.O. & Larchenko, O.V. (2021). Modeli i metody intelektual'noho analizu danykh: navchal'nyy posibnyk [Models and Methods of Intelligent Data Analysis: Tutorial]. Kherson: Vyshemyrs'kyy V.S. ISBN 978-617-7941-60-5 [in Ukrainian].

5. Moor, Jeffrey H. & Weatherford, Larry R. (2001). Decision Modeling with Microsoft Excel. Upper Sddle River, New Jersey 07458: Published by Prentice Hall, Inc. ISBN 0-13-017789-X.

**UDC 004.62**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-07**

# ENHANCING DISTRIBUTED STREAM PROCESSING THROUGH ARTIFICIAL INTELLIGENCE: A REVIEW AND PROPOSAL FOR LSTM INTEGRATION

***Ilin M.O.,*** *PhD student, Department of Computer Systems Software, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine https://orcid.org/0009-0001-0803-3726, max.ilin@yahoo.com*

***Oleshchenko L.M.,*** *Candidate of Technical Sciences, Associate Professor at the Computer Systems Software Department, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine https://orcid.org/0000-0001-9908-7422, oleshchenkoliubov@gmail.com*

***Abstract.*** *The proliferation of Internet of Things (IoT) devices and sensors is generating massive volumes of real-time, heterogeneous data streams. Organizations need to be able to integrate and analyze these high-velocity, multi-formatted data to extract value. However, traditional batch analytics paradigms struggle with the volume and latency demands. This paper reviews a distributed stream data processing architecture designed to address these challenges. The proposed information system provides a pipeline for ingesting high-volume, heterogeneous data streams; automatically homogenizing the data into a consistent schema; and enabling Complex Event Processing (CEP) to detect situations of interest through real-time pattern matching. The feasibility of the architecture was demonstrated by implementation in a smart water management system to process and analyze real-time meter data. However, further enhancements using artificial intelligence (AI) technologies could improve the efficiency, accuracy, and adaptability of the system. Specifically, this paper hypothesizes integrating Long Short-Term Memory (LSTM) neural networks to add predictive analytics capabilities. The LSTM models would forecast expected values from historical data, identify deviations, and augment the stream events with predictions. This would enable the CEP engine to operate at a higher level detecting pattern anomalies rather than individual outliers. The AI integration is projected to increase the accuracy of anomaly detection based on more holistic data context. However, sizable research is still required to prototype, implement, and validate the fusion of stream processing with recurrent neural networks. This paper provides an initial perspective on the future convergence of big data pipelines and artificial intelligence to support rapidly evolving, mission-critical streaming applications in IoT domains. It lays the foundation for an intelligent distributed architecture that can handle heterogeneous data at scale.*

***Key words:*** *software methods, heterogeneous streaming data, Python, data processing information systems, distributed computing, AI technologies, LSTM, CEP.*

# ОСОБЛИВОСТІ ОБРОБКИ ПОТОКОВИХ ДАНИХ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ: ОГЛЯД І ПОКРАЩЕННЯ ІСНУЮЧИХ ПІДХОДІВ ЗА ДОПОМОГОЮ ІНТЕГРАЦІЇ LSTM

***Максим Ільїн,*** *аспірант кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна https://orcid.org/0009-0001-0803-3726, max.ilin@yahoo.com*

***Любов Олещенко,*** *канд. техн. наук, доцент кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна https://orcid.org/0000-0001-9908-7422, oleshchenkoliubov@gmail.com*

***Анотація.*** *Поширення пристроїв і датчиків інтернету речей сприяє генерації величезних об'ємів гетерогенних потоків даних у реальному часі. Організації повинні мати можливість інтегрувати й аналізувати потокові дані різного формату для отримання бізнес-цінності. Однак традиційні парадигми пакетної аналітики мають проблеми з об'ємом даних і затримкою в часі. У статті розглядається архітектура розподіленої потокової обробки даних, спрямована на вирішення цих проблем. Запропонована інформаційна система забезпечує конвеєр для приймання великого обсягу неоднорідних потоків даних, автоматичного упорядкування даних у послідовну схему, а також увімкнення комплексної обробки подій для виявлення специфічних ситуацій за допо-*

*могою зіставлення шаблонів у реальному часі. Здійсненність архітектури було продемонстровано впровадженням в інтелектуальну систему управління водними ресурсами для обробки й аналізу даних лічильників у реальному часі. Подальші вдосконалення за допомогою технологій штучного інтелекту (ШІ) можуть підвищити ефективність, точність і адаптивність інформаційної системи. Зокрема, у статті висувається гіпотеза про інтеграцію нейронних мереж довготривалої короткочасної пам'яті (LSTM) для додавання можливостей прогнозної аналітики. Моделі LSTM дають змогу прогнозувати очікувані значення на основі історичних даних, визначати відхилення та доповнювати потоки подій прогнозами. Завдяки цьому механізм комплексної обробки подій працюватиме на вищому рівні, виявляючи типові аномалії, а не окремі викиди. Очікується, що інтеграція ШІ підвищить точність виявлення аномалій на основі більш цілісного контексту даних. На сьогодні відсутні дослідження для прототипування, реалізації та перевірки інтеграції рекурентних нейронних мереж у потокову обробку даних. У роботі представлено початковий погляд на майбутню конвергенцію конвеєрів великих даних і штучного інтелекту для підтримки критично важливих програм потокового передавання даних у доменах інтернету речей.*

***Ключові слова:*** *програмні методи, гетерогенні потокові дані, Python, інформаційні системи оброблення даних, розподілені обчислення, технології штучного інтелекту, LSTM, CEP.*

**Introduction**

Processing and analyzing real-time data streams is an increasingly important capability in domains ranging from IoT to finance. However, integrating and making sense of heterogeneous data formats poses challenges. In 2020 Corral-Plaza et al. proposed a novel stream processing architecture to address these issues. The paper [1] proposes a novel architecture for processing streaming data from diverse Internet of Things (IoT) devices. The architecture enables real-time data integration, filtering, and analytics for various applications, including predictive maintenance, quality control, and environmental monitoring. The authors evaluate the architecture using several case studies and demonstrate its effectiveness in handling complex data streams with varying levels of accuracy and latency.

Today, there are no studies for prototyping, implementation and verification of the integration of recurrent neural networks in streaming data processing, so this problem is relevant and requires further research.

This paper reviews distributed information system in detail, including its layers, components, and sample implementations. Building on this foundation, potential augmentations using artificial intelligence are explored to enhance real-time data analytics. Finally, conclusions are presented on the feasibility and projected improvements from integrating AI into stream processing systems.

**Research**

Today, distributed information systems have been developed for stream data processing of various applications. The research paper by Hu et al. (2016) proposes a information system for sensor data processing from multiple sources in real-time. The system is designed to handle heterogeneous sensor data and can be used in various applications such as environmental monitoring, health monitoring, and traffic management. The authors present a novel stream processing algorithm that can handle data from different sources with different data formats and characteristics [2]. The paper Montori F. et al. (2016) discusses the challenges and opportunities of integrating multiple data sources in the context of the Internet of Things (IoT). The authors propose a framework for integrating heterogeneous data sources, including sensor data, social media, and context-aware systems. They discuss the importance of data quality and security in this integration and highlight the potential benefits of collaborative data sharing for IoT applications [3]. The authors of the article by Nadal S. et al. in 2017 proposed a framework for building semantic-aware Big Data systems. The authors present a software reference architecture that integrates semantic technologies with Big Data processing to enable more efficient and effective data analysis. The architecture consists of several components, including a semantic data lake, a semantic metadata management system, and a set of semantic-aware data processing modules. The paper provides a detailed description of each component and their interactions, as well as use cases that demonstrate the effectiveness of the architecture [4]. The research paper by D'Silva et al. (2017) proposes a system for real-time processing of IoT events and historic data using Apache Kafka and Apache Spark with the Dashing framework. The system enables efficient processing and analysis of large datasets from IoT devices, enabling real-time insights and decision-making [5]. The article by Estévez-Ayres I. et al. (2017) proposes a novel infrastructure called Lostrego, designed to streamline the collection and analysis of diverse educational data in real-time. Lostrego adopts a distributed architecture, allowing for efficient processing and scalability, and supports diverse data sources, including learning management systems, student information systems, and online collaboration platforms. The research demonstrates the effectiveness of Lostrego through a series of case studies, showcasing its ability to provide real-time insights into educational processes and outcomes [6]. The research by Traore et al. (2016) explores the use of artificial neural networks (ANNs) for short-term forecasting of evapotranspiration (ET) using public weather forecast restricted messages. The authors use a dataset from the southeastern United States and demonstrate the potential of ANNs in improving ET forecasts. The study contributes to the development of accurate and reliable ET forecasting methods, which are essential for water resource management and crop planning [7]. The by Greff et al. (2017) investigates the search space of Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN) widely used in Natural Language Processing (NLP) tasks. The authors propose a novel approach to explore

the vast search space of LSTM architectures, enabling the identification of optimal hyperparameters and evaluating the performance of LSTM models [8]. The paper by Karim F., Majumdar S., Darabi H., and Chen S. (2018) proposes a novel deep learning architecture that combines the strengths of Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNNs) for time series classification. The authors introduce the LSTM-FCN model, which utilizes fully convolutional layers to improve the feature extraction process and reduce computational complexity. The paper provides a detailed evaluation of the LSTM-FCN model on several benchmark datasets, demonstrating its superior performance compared to traditional time series classification methods [9].

Corral-Plaza et al. proposed an architecture for processing and analyzing heterogeneous data streams in real-time from Internet of Things (IoT) devices.

The information system is organized into three layers: data producers, data processing and data consumers.

On the data producers layer IoT devices and sensors generate heterogeneous data streams. The data can be structured (e.g. JSON, XML) or unstructured (e.g. CSV files).

Data processing layer contains two main components – a stream processing platform (e.g. Kafka Streams) and a complex event processing (CEP) engine (e.g. Esper). The stream processing platform consumes the heterogeneous data streams, homogenizes the data into a common format, and generates simple events. The CEP engine analyzes the simple events to detect situations of interest.

On the data consumers layer endpoints like databases, smartphones, or web services consume the processed data and detected situations from the system.

**Architecture**

The data processing layer contains the main components for homogenizing and analyzing the heterogeneous data streams:

*Kafka Cluster*

A Kafka cluster stores the incoming heterogeneous data streams in Kafka topics.

Data producers publish messages to input topics, while consumers can subscribe to output topics (Fig. 1). Kafka provides high throughput and scalability [10].

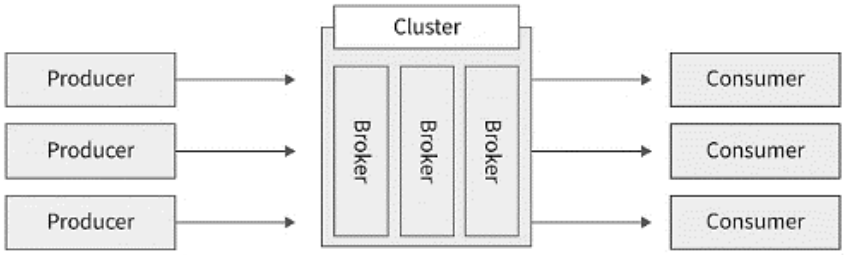


Fig. 1. Kafka cluster structure

In the water management implementation, there are:

- an input topic for sensor data from smart water meters;
- an output topic for detected anomalies and alerts.

*Kafka Streams Application*

The Kafka Streams application consumes messages from the input topic, homogenizes the data, and generates simple events for analysis (Fig. 2).

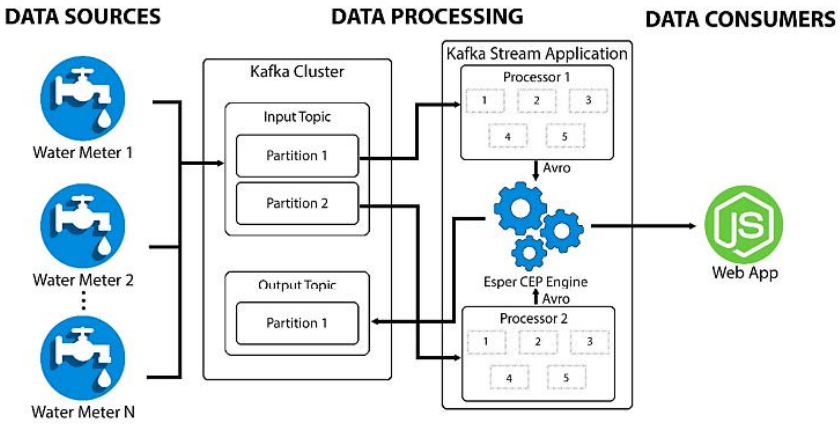


Fig. 2. Software architecture adapted to the water management scenario [1]

Key steps to use the information system are next.

1. Consume data from the input topic partitions.
2. Detect data format (JSON, XML, CSV).
3. Homogenize data into a common JSON format.
4. Generate Avro schema for the data.
5. Create Avro simple events for analysis.

*Esper CEP Engine*

The Esper CEP engine receives the Avro simple events from Kafka Streams. It allows defining EPL patterns that match event conditions and detect situations of interest. When a pattern matches, a new complex event is generated and published to the output topic. For example, anomalies like water meter reading errors can trigger alerts. New patterns can also be deployed at runtime without stopping the system.

***Implementation Example***

Corral-Plaza et al. implemented the architecture for a smart water supply network. Heterogeneous data from smart water meters was processed and analyzed. For example, heterogeneous sensor data from smart water meters is consumed and homogenized into a JSON format with a schema (Table 1). This generates simple "WaterMeasurement" events for analysis.

Table 1

Sample sensor data in JSON format

```
{
  "serialNumber": "A87SAA77188",
  "dateTime": "2018-07-06T05:00:00.000Z",
  "volumeM3": 5.4366397,
  ...
}
```

The Kafka Streams application homogenizes the sensor data into Avro "WaterMeasurement" events (Table 2).

Table 2

Avro "WaterMeasurement" events in Java

```
class HomogenizeSensorData {
  @StreamListener("input-topic")
  @SendTo("cep-topic")
  public WaterMeasurement process(String message) {
    // Detect format
    if (isJSON(message)) {
      // Deserialize JSON
    } else if (isCSV(message)) {
      // Parse CSV
    }
    // Homogenize and create Avro event
    WaterMeasurement event = ...;
       return event;
  }
}
```

The Esper CEP engine detects anomalies using EPL patterns (Table 3).

Table 3

Example of an EPL pattern, SQL

```
INSERT INTO WaterMeterAnomaly
SELECT *
FROM WaterMeasurement
WHERE volumeM3 < 0
```

Overall, the architecture provided real-time processing of heterogeneous data streams in a water management IoT system.

The main advantages of the architecture are:

- real-time processing of high volumes of heterogeneous data streams;
- automatic homogenization of data into a common format;
- real-time analysis and detection of situations of interest;
- high scalability by adding more stream processing nodes.

***Proposed AI Integration Method***

Long Short-Term Memory (LSTM) neural networks are a type of recurrent neural network (RNN) that have proven to be effective for processing and forecasting streaming data, particularly in time series analysis. LSTMs are designed to handle sequences of data. They are capable of processing data points one at a time in a sequential manner, making them well-suited for streaming data where the order of the data points matters. LSTMs can capture long-term dependencies in the data, which is important for understanding patterns and trends in time series data. Unlike traditional feedforward neural networks, LSTMs can remember information from much earlier in the sequence. LSTMs can be adapted to work with streaming data in an online learning setup. As new data points arrive, the LSTM can be updated and retrained in real-time, allowing it to adapt to changing patterns and trends. LSTMs can handle sequences of varying lengths. In a streaming data scenario, the length of the data sequence may change over time, and LSTMs are flexible enough to handle this. LSTMs can be used in a stateful mode, where the internal state is carried over from one batch of data to the next. This is particularly useful for streaming data, as it allows the network to maintain context across multiple time steps. LSTMs can automatically learn relevant features from the data, reducing the need for manual feature engineering. This is beneficial when dealing with streaming data, where the data distribution may change over time. LSTMs can handle multivariate time series data, which is common in many streaming data applications. They can model dependencies between multiple variables simultaneously. LSTMs can be used for anomaly detection in streaming data. By modeling the normal behavior of a system, they can detect deviations from this normal behavior in real-time. LSTMs are often used for time series forecasting. They can be trained to predict future data points in a sequence, which is valuable for making predictions in streaming data applications. LSTMs can be integrated into systems for real-time decision making. For example, in finance, they can be used to make trading decisions based on streaming market data.

The architecture of the recurrent neural network is shown in Fig. 3.

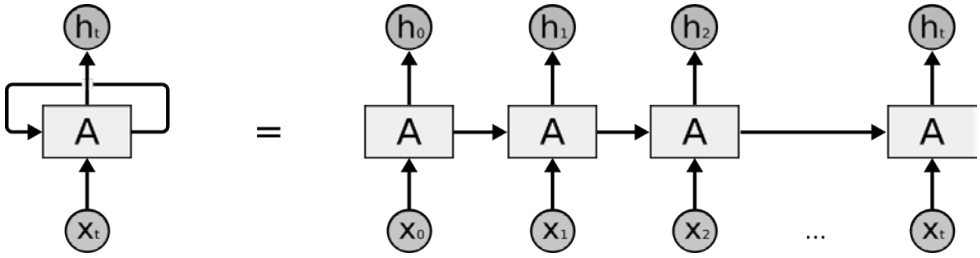


Fig. 3. Recurrent neural network in sweep

This network is particularly adept at forecasting time series data due to its ability to factor in past experience. The key distinction between LSTM and traditional RNN lies in the inclusion of specialized nodes dedicated to retaining information over extended time intervals. This is achieved by these nodes abstaining from using the activation function, ensuring that data remains unchanged as it's stored for future reference.

LSTM takes the form of a chain of recurrent neural network modules. Fig. 4 shows an LSTM structure that has four hidden layers and uses a sigmoid function and a hyperbolic tangent as the activation function.

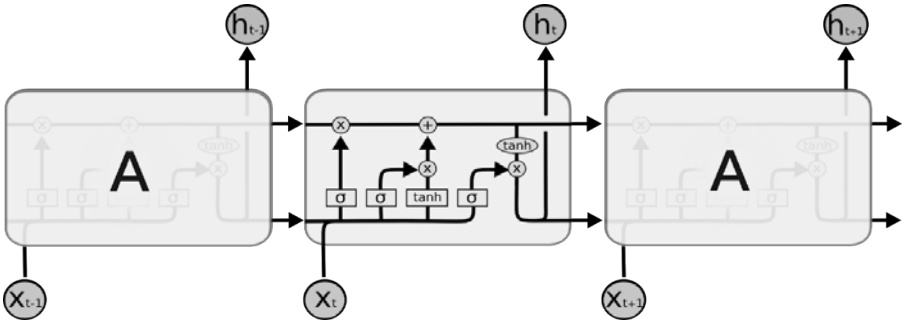


Fig. 4. LSTM structure

The neural network is trained using backpropagation in time, for which iterative gradient descent is used, which changes each weight coefficient in proportion to its derivative with respect to the error. Traditional LSTM with forgetting gates $c_0 = 0$, $h_0 = 0$:

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (1)$$

$$i_t = \sigma_g\left(W_i x_t + U_i h_{t-1} + b_i\right) \quad (2)$$

$$o_t = \sigma_g\left(W_o x_t + U_o h_{t-1} + b_o\right) \quad (3)$$

$$c_t = f_t c_{t-1} + i_t \sigma_c\left(W_c x_i + U_c h_{t-1} + b_c\right) \quad (4)$$

$$h_t = o_t \sigma_h\left(c_t\right) \quad (5)$$

Variables: $x_t$ is input vector; $h_t$ – output vector; $c_t$ – vector of states; W, U, and b are parameter matrices and a vector; $f_t$, $i_t$ and $o_t$ – gate vectors: $f_t$ – the forgetting gate, the weight of remembering old information, $i_t$ – the input gate, the weight of receiving new information, $o_t$ – the output gate, the candidate for output (Fig. 5). Activation functions: $\sigma_g$ – based on the sigmoid, $\sigma_c$ – based on the hyperbolic tangent, $\sigma_h$ – based on the hyperbolic tangent, in this work it is assumed that $\sigma_h(x) = x$.

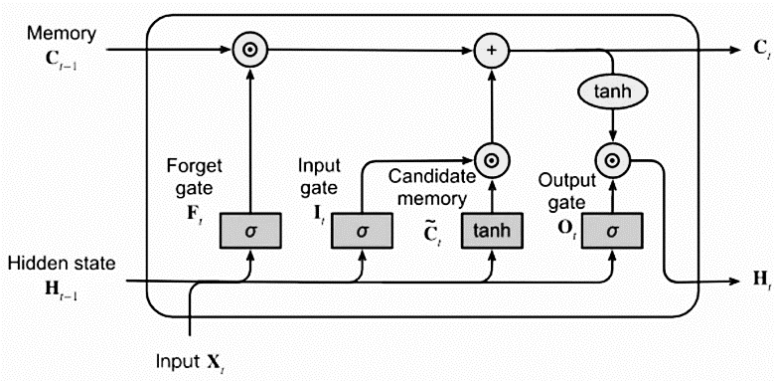


Fig. 5. A simple LSTM block with three gates: input, output and forget

To enable more efficient and intelligent processing, a predictive analytics module based on LSTM neural networks could be added between the Kafka Streams application and the Esper CEP engine [11].

The LSTM model would take the homogenized data streams from Kafka Streams as input and make multi-step predictions about expected values. For example, in the water management use case it could forecast anticipated consumption and detect when actual readings diverge significantly from the prediction.

The LSTM module would have the following components:

- Data preparation – the Avro event streams are formatted as sequences for time series prediction.
- LSTM model – Long Short-Term Memory network is trained on historical data to make multi-step forward predictions.
- Anomaly detection – errors between predicted values and actual events are calculated. Significant deviations are flagged as anomalies.
- Integration – the predicted values and anomalies are appended to the Avro events emitted to the CEP engine.

Table 4

Pseudocode for proposed AI integration

```
DataPreparation(AvroStream):
    format avro data as time series
  LSTMModel(HistoricalData):
    train LSTM network
    return model
  Predict(Model, AvroStream):
    make multi-step predictions
    calculate prediction errors
    flag anomalies
  AugmentAvroStream(AvroStream, Predictions, Anomalies):
    add predictions and anomalies to Avro events
    return AugmentedStream
```

Adding predictive abilities would allow the CEP engine to operate on a higher level, detecting pattern anomalies rather than individual outliers. This is expected to improve the accuracy of anomaly detection.

The LSTM module would run continuously alongside Kafka Streams and incrementally update its model on new data. This would allow the predictions to adapt to evolving conditions. It's important to note that while LSTMs have many advantages for processing and forecasting streaming data, they are not without

challenges. They can be computationally expensive, and training them with large datasets in real-time may require specialized hardware or distributed computing. Additionally, choosing the right architecture and hyperparameters for specific streaming data application is crucial to achieve the best performance (Fig. 6).

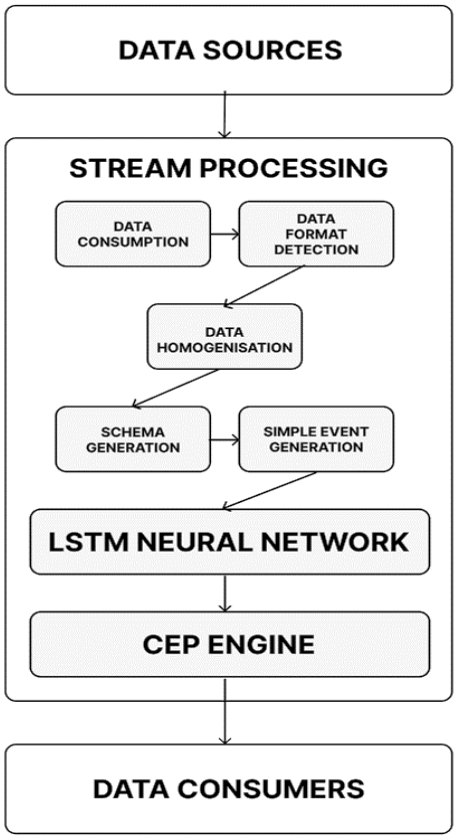


Fig. 6. Proposed architecture enhanced with AI usage

Overall, integrating recurrent neural networks could significantly improve the intelligence of the distributed stream processing system. While requiring further research, this AI-based approach has potential for making heterogeneous data analytics more efficient.

**Conclusions**

In this paper, we reviewed a distributed system stream processing system architecture for handling heterogeneous data that provides an effective pipeline for ingesting high-volume, real-time streams of heterogeneous data formats, automatically homogenizing the data into a consistent schema and enabling complex event processing for real-time pattern detection and alerts.

The feasibility of the approach was demonstrated through an implementation in a smart water management IoT system. However, further enhancements using artificial intelligence could improve the efficiency, accuracy, and adaptability of the system.

Specifically, we proposed integrating LSTM neural networks to add predictive analytics capabilities. By forecasting expected values and preemptively identifying anomalies, this AI integration could potentially improve the anomaly detection accuracy. While promising, this AI approach requires further research into production deployments, model training requirements, and integration architectures. As next steps, a proof-of-concept prototype should be implemented to quantify the concrete benefits for real-world streaming applications.

In the long-term, blending stream processing with artificial intelligence will be essential to handling accelerating volumes of heterogeneous and complex data. This paper presented initial perspectives on accomplishing that fusion through a distributed, real-time analytics pipeline augmented with machine learning. There remains ample open challenges and opportunities for future innovations in this domain.

***References***

1. Corral-Plaza, D., Medina-Bulo, I., Ortiz, G., & Boubeta-Puig, J. (2020). A stream processing architecture for heterogeneous data sources in the Internet of Things. *Computer Standards & Interfaces*. 70 (2020) 103426. DOI: 10.1016/j.csi.2020.103426.

2. Hu, L., Sun, R., Wang, F., Fei, X., Zhao, K. (2016). A stream processing system for multisource heterogeneous sensor data, *Sens.*, https://doi.org/10.1155/2016/4287834.

3. Montori, F., Bedogni, L., Bononi, L. (2016). On the integration of heterogeneous data sources for the collaborative Internet of Things, *IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI)*, 2016, pp. 1–6, https://doi.org/10.1109/RTSI.2016.7740616.

4. Nadal, S., Herrero, V., Romero, O., Abelló, A., Franch, X., Vansummeren, S., Valerio, D. (2017). A software reference architecture for semantic-aware Big Data systems, *Inf. Softw. Technol.* 90. 75–92, https://doi.org/10.1016/j.infsof.2017.06.001.

5. D'silva, G.M., Khan, A., Gaurav, S.B. (2017). Real-time processing of IoT events with historic data using apache Kafka and Apache Spark with dashing framework, *2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pp. 1804–1809, https://doi.org/10.1109/RTEICT.2017.8256910.

6. Estévez-Ayres, I., Arias Fisteus, J., Delgado-Kloos, C. (2017). Lostrego: a distributed streambased infrastructure for the real-time gathering and analysis of heterogeneous educational data, *Netw. Comput. Appl*. 100, pp. 56–68, https://doi.org/10.1016/j.jnca.2017.10.014.

7. Traore, S., Luo, Y.F., Fipps, G. (2016). Deployment of artificial neural network for short-term forecasting of evapotranspiration using public weather forecast restricted messages. *Agricultural Water Management* 163 (1), 363–379. https://doi.org/10.1016/j.agwat.2015.10.009.

8. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R. and Schmidhuber, J. (2017). LSTM: A Search Space Odyssey, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, doi: 10.1109/TNNLS.2016.2582924.

9. Karim, F., Majumdar, S., Darabi, H. and Chen, S. (2018). LSTM Fully Convolutional Networks for Time Series Classification, *IEEE Access*, vol. 6, pp. 1662–1669, doi: 10.1109/ACCESS.2017.2779939.

10. Stojanović Natalija, and Dragan Stojanović. (2020). Big mobility data analytics for traffic monitoring and control. *Facta Universitatis, Series: Automatic Control and Robotics* 19, no. 2: 087. http://dx.doi.org/10.22190/fuacr2002087s.

11. Cheng H., Xie Z., Wu L. et al. (2019). Data prediction model in wireless sensor networks based on bidirectional LSTM. *Wireless Com Network* 2019, 203. https://doi.org/10.1186/s13638-019-1511-4.

**УДК 511**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-08**

# МНОЖИНА ЦІЛИХ ЧИСЕЛ З УНІКАЛЬНИМИ СУМАМИ ДВОХ ЕЛЕМЕНТІВ

***Павленко О. Ю.,*** *Відкритий міжнародний університет розвитку людини «Україна», Інститут комп'ютерних технологій, Київ, Україна*
*https://orcid.org/0000-0001-9690-5987, asd97456@gmail.com*

***Тимошенко А. Г.,*** *канд. техн. наук, Відкритий міжнародний університет розвитку людини «Україна», Інститут комп'ютерних технологій, Київ, Україна*
*https://orcid.org/0000-0003-0954-3186, timoshag@i.ua*

***Тимошенко О. М.,*** *канд. фіз.-мат. наук, Відкритий міжнародний університет розвитку людини «Україна», Інститут комп'ютерних технологій, Київ, Україна*
*https://orcid.org/0009-0003-4331-1706, omtimoshenko@gmail.com*

***Анотація.*** *Розглянуто задачу створення множини натуральних чисел з унікальними сумами будь-яких двох елементів. Як елементи можна використовувати натуральний ряд (1, 2, 3, …, n), або починати з нуля (0, 1, 2, …, n). У першому випадку під парними сумами розуміємо суму двох будь-яких елементів. Наприклад, для 4 елементів множини отриманий ряд (1, 2, 3, 5) має такі всі можливі унікальні парні суми (1 + 2 = 3; 1 + 3 = 4; 1 + 5 = 6; 2 + 3 = 5; 2 + 5 = 7; 3 + 5 = 8). У другому випадку кожен елемент ряду може розглядатися як парна сума з нулем. Тоді для 4 елементів множини (0, 1, 2, 4) маємо всі можливі унікальні парні суми (0 + 1 = 1; 0 + 2 = 2; 0 + 4 = 4; 1 + 2 = 3; 1 + 4 = 5; 2 + 4 = 6). Тобто в цьому випадку сума двох значущих елементів не повинна збігатися з жодним елементом множини й іншими парними сумами. На цих прикладах ми бачимо, що це різні задачі, і в роботі розглядається другий варіант.*

*Запропоновано алгоритм пошуку множини натуральних чисел за різних значень кількості елементів. Як критерії оптимальної побудови ряду використовується мінімум суми елементів і мінімальне значення останнього елемента. Наведено алгоритм і програма створення такого ряду. Отримано результат для 8 елементів множини. Програма написана алгоритмічною мовою програмування Python. Вона має компактний характер завдяки використанню бібліотеки itertools. Результати досліджень наведені в таблиці для різних значень кількості значущих елементів множини. Для найбільшої кількості елементів отримано кілька результатів. Це пов'язано з вибором рядів за двома критеріями. У наведених результатах досліджень ці критерії збігаються за кількості елементів n < 8. Для восьми елементів множини отримано різні набори натуральних чисел для кожного мінімального критерія.*

***Ключові слова:*** *унікальні суми множини натуральних чисел, адитивно-різний ряд, парні суми.*

# SET OF INTEGERS WITH UNIQUE SUMS OF TWO ELEMENTS

***Oleh Pavlenko,*** *Institute Open International University of Human Development "Ukraine", Kyiv, Ukraine*
*https://orcid.org/0000-0001-9690-5987, asd97456@gmail.com*

***Anatolii Tymoshenko,*** *Ph.D., Institute Open International University of Human Development "Ukraine", Kyiv, Ukraine*
*https://orcid.org/0000-0003-0954-3186, timoshag@i.ua*

***Oksana Tymoshenko,*** *Ph.D., Institute Open International University of Human Development "Ukraine", Kyiv, Ukraine*
*https://orcid.org/0009-0003-4331-1706, omtimoshenko@gmail.com*

***Abstract.*** *The problem of creating a set of natural numbers with unique sums of any two elements is considered. The elements can be taken from the natural series (1, 2… n) or start from zero (0, 1, 2… n). In the first case, even sums necessarily mean the sum of two elements. For example, for a set of 4 elements, the obtained series (1, 2, 3, 5) has the following unique even sums (1 + 2 = 3; 1 + 3 = 4; 1 + 5 = 6; 2 + 3 = 5; 2 + 5 = 7; 3 + 5 = 8). In the second case, each element of the series can be considered as an even sum with zero. Then, for a set of 4 elements (0, 1, 2, 4), we have all possible unique even sums (0 + 1 = 1; 0 + 2 = 2; 0 + 4 = 4; 1 + 2 = 3; 1 + 4 = 5; 2 + 4 = 6). Thus, in this case, the sum of two significant elements should not coincide with any element of the set or with other even sums. These examples illustrate that these are different problems, and the work focuses on the second variant. The proposed algorithm for finding a set of natural numbers for different*

*numbers of elements is discussed. The criteria for optimal series construction include the minimum sum of elements and the minimum value of the last element. The algorithm and program for creating such a series are provided. The obtained result is for 8 elements of the set. The program is written in the Python programming language using the itertools library, which makes it compact. The research results are presented in a table for various numbers of significant elements in the set. For the largest number of elements, several results are obtained, as the series are chosen based on two criteria. In the presented research results, these criteria coincide for numbers of elements n < 8. For eight elements of the set, different sets of natural numbers are obtained for each minimal criterion.*

***Key words:*** *unique sums of natural numbers set, additive-different series, even sums.*

**Вступ**

Під адитивно-різним рядом ми розуміємо множину натуральних чисел, у яких суми підмножин мають унікальні значення. Таку властивість має ряд $A^k$, де $A$ – елементи натурального ряду і $A > 1$, а $к = 0, 1, 2, ..., n$. Найпростіший ряд $2^n$. Ця властивість випливає з того, що максимальний елемент ряду на одиницю більший за суму всіх попередніх елементів. Інколи виникає проблема побудови ряду, де максимальний елемент якомога менший. Уперше такий ряд був запропонований у роботі [1]. Більш досконалий ряд запропоновано в роботі [2]. Результати створення таких рядів наведені в таблиці 1. Там доведено неможливість вдосконалити побудову ряду з меншим максимальних елементом. Це було програмно підтверджено в роботі [3]. Використання абсолютно адитивно-різних чисел знайшло застосування у розв'язані задачі комівояжера [4].

Крім ряду пошуку оптимального максимального елемента існують задачі пошуку ряду з унікальними сумами в заданій послідовності. Тут побудова ряду досягається перебором, оскільки неможливо використовувати попередній ряд. У роботі [5] було знайдено оптимальний ряд для 9 чисел. Побудова більшого ряду залишається відкритою задачею.

У роботі розглядається задача створення ряду адитивно-різних чисел з унікальними парними сумами.

Таблиця 1

Ряд абсолютно адитивно-різних чисел

| n | Ряд абсолютно адитивно-різних чисел | | | | | | | | max | Sum |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | 1 | 1 | **1** |
| 2 | | | | | | | 1 | **2** | 2 | **3** |
| 3 | | | | | | 2 | **3** | 4 | 4 | **9** |
| 4 | | | | | 3 | 5 | **6** | 7 | 7 | **21** |
| 5 | | | | 6 | 9 | **11** | 12 | 13 | 13 | **51** |
| 6 | | | 11 | 17 | **20** | 22 | 23 | 24 | 24 | **117** |
| 7 | | 20 | 31 | 37 | **40** | 42 | 43 | 44 | 44 | **257** |
| 8 | 40 | 60 | 71 | **77** | 80 | 82 | 83 | 84 | 84 | **577** |

Виділенні клітинки використовуються для побудови наступного ряду. Для цього використовується попередній ряд.

**Пошук множини натуральних чисел з унікальними парними сумами**

Тут маємо два критерії пошуку ряду. Ряд з мінімальним останнім елементом, або з мінімальною сумою всіх елементів ряду. За певної кількості елементів ці критерії можуть збігатися, тобто ми отримуємо ряд з мінімальним останнім елементом і найменшою сумою. В інших випадках одна послідовність має менший останній елемент, але більшу суму.

Варто визначити набір елементів множини ряду. Зазвичай ми використовуємо натуральний ряд або натуральний ряд і нуль. У другому випадку всі суми двох елементів не можуть збігатися з будь-яким елементом ряду. Ми розглядаємо другий випадок.

У роботі пропонується розв'язувати задачу за обома критеріями, тобто визначення мінімального останнього елемента та визначення мінімальної суми ряду.

Задаємо граничний діапазон чисел, у якому будемо визначати елементи ряду. З попередніх досліджень це значення для ряду до 7 чисел було вибрано діапазон 1…40. Збільшення діапазону суттєво впливає на час розрахунку.

З бібліотеки комбінаторики *itertools* використовується функція combinations(iterable, [r]) – створення комбінації довжиною r з iterable без повторюваних елементів. Приклад combinations('ABCD', 2) --> AB AC AD BC BD CD

***Алгоритм роботи програми***

Вхідні змінні:

a – масив елементів;

n = 8 число елементів;

maxd – граничне значення діапазону елементів;
adsum – граничне значення межі сум ряду;
admax – останній елемент ряду.
Генерація масиву a:
a – масив від 1 до maxd.
Генеруються всі можливі комбінації рядів від 1 до 40 з довжиною n без повторюваних елементів.

***Перевірка на унікальність сум***

Знаходимо суми всіх комбінацій: масив елементів плюс масив сум комбінацій двох елементів. Об'єднуємо їх в один масив. Далі перевіряємо наявність у масиві однакових сум чисел.
if len(adetcheck) == len(set(adetcheck)),
len – довжина масиву,
set – видалення дублів в масиві.
Порівнюємо довжини двох масивів. Якщо довжини збігаються, тобто немає однакових сум, то це знайдений ряд, що має унікальні суми будь-яких двох чисел.
Також йде перевірка суми знайденого ряду із сумою попереднього ряду. З них залишаємо ряд із меншою сумою. Результат виводимо на екран.
Друга перевірка – перевірка на максимальний елемент знайденого ряду з аналогічним попереднього ряду, з яких залишаємо ряд із меншим останнім елементом. Результат також виводимо на екран.
Програма 1. Пошук адитивно-різних рядів з унікальними парними сумами.

```
%%time
import numpy as np
from itertools import *
a = []
n = 8
maxchislo = 41
adsum = 50000000
admax = 50000000

for r in range(1,maxchislo):
    a.append(r)
#Перевірка на адитивність. Суми одних чисел + суми двох чисел ряду.
for j in combinations(a, n):
    adetcheck = []
    adetcheck = [sum(i) for i in combinations(j, 1)] + [sum(i) for i in combinations(j, 2)]
  #Перевірка адитивності ряду на співпадіння сум.
    if len(adetcheck) == len(set(adetcheck)) and sum(j)<=adsum:
        adsum=sum(j)
         # Вивід максимально-мінімального числа ряду за знайдену раніше
        print(j,' suma= ',sum(j))
    if len(adetcheck) == len(set(adetcheck)) and max(j)<=admax:
        admax=max(j)
# Вивід мінімальної суми ряду за знайдену раніше
        print(j,' max= ',max(j),' suma= ',sum(j))
```

**Результати досліджень**

***Результати пошуку ряду***

За допомогою програми проведені дослідження пошуку елементів ряду для кількості n = 1, 2, 3, …, 8. Отримані результати наведені в таблиці 2.

Таблиця 2

Множини натуральних чисел від 1 до 8 елементів

| n | Адитивно-різний ряд парних чисел | | | | | | | | max | sum | time | maxd |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | 1 | 1 | 1 | 0 | 40 |
| 2 | | | | | | | 1 | 2 | 2 | 3 | 3 ms | 40 |
| 3 | | | | | | 1 | 2 | 4 | 4 | 7 | 37.8 ms | 40 |
| 4 | | | | | 1 | 2 | 4 | 7 | 7 | 14 | 385 ms | 40 |
| 5 | | | | 1 | 2 | 4 | 7 | 12 | 12 | 26 | 3.13 s | 40 |
| 6 | | | 1 | 2 | 4 | 8 | 13 | 18 | 18 | 46 | 22.1 s | 40 |
| 7 | | 1 | 2 | 4 | 8 | 14 | 19 | 24 | 24 | 72 | 2 min 8 s | 40 |
| 8 | 1 | 2 | 4 | 8 | 14 | 19 | 24 | 40 | 40 | **112** | 12 min 7 s | 41 |
| 8 | 1 | 2 | 4 | 8 | 15 | 24 | 29 | 34 | **34** | 117 | 12 min 7 s | 41 |

*Примітки: n – кількість елементів ряду, max – максимальне значення ряду, sum – сума ряду, time – час виконання програми, maxd – граничне значення діапазону пошуку елементів.*

Наведені результати роботи програми отримані на комп'ютері з такими параметрами:
Inter® Core™ i-7 2600 CPU @ 3,40GHz, ОЗУ 6 ГБ, Windows 10.

**Висновки**

На відміну від абсолютно адитивно-різних чисел, не існує простої формули побудови нового ряду на базі попереднього. Тому фактично ми ведемо пошук серед всіх можливих комбінацій. При цьому час пошуку для нового набору суттєво зростає. Із цієї причини ми отримали оптимальні результати для набору рядів до 8 елементів.

Отримані ряди дають змогу стверджувати, що результати носять оптимальний характер, тобто не існують менші суми набору елементів і менший максимальний елемент ряду.

Для пошуку рядів з більшою кількістю елементів потрібно використовувати інші алгоритми. Можливо, слід використати багатопроцесорні комплекси.

***Література***

1. Алоян Г. С., Тимошенко А. Г. Спосіб побудови ряду адитивно-різних чисел. Кібернетика. 1969. № 6. С. 64–66.

2. Тимошенко А. Г., Лузан К. О., Рогачова Т. В. Створення ряду цілих чисел з унікальними сумами елементів. Наукові записки НаУКМА. 2002. Том 20. Спеціальний випуск у двох частинах. Частина 2. С. 516–517.

3. Pavlenko O.Yu. Determination of the optimal sequence of additively distinct numbers., EUROPEAN SCIENTIFIC DISCUSSIONS. Proceedings of X International Scientific and Practical Conference Rome, Italy, 2021, August 15–17, p. 75–79.

4. Oleh Pavlenko, Anatolii Tymoshenko, Igor Melnyk, Andriy Luntovskyy et al. Searching Extreme Paths Based on Travelling Salesman's Problem for Wireless Emerging Networking, in Springer LNEE 965. Emerging Networking in the Digital Transformation Age: Approaches, Protocols, Platforms, Best Practices, and Energy Efficiency, by Springer LNEE 2023, issue 965, Springer Nature Cham, 24 p. (chapter #16).

5. Павленко О. Ю., Тимошенко А. Г., Бондаренко В. М. Алгоритм пошуку ряду послідовно адитивно-різних чисел. Вісник Університету «Україна» Інформатика обчислювальна техніка та кібернетика. 2017, № 1(18). С. 118–121.

***References***

1. Aloian, H.S., Timoshenko, A.H. (1969). Method of constructing a series of additively distinct numbers. Cybernetics. № 6. P. 64–66.

2. Timoshenko, A.H., Luzan, K.O., Rogachova, T.V. (2002). Creation of a series of integers with unique sums of elements. Scientific Notes of NaUKMA. Vol. 20. Special issue in two parts. Part 2. P. 516–517.

3. Pavlenko O.Yu. Determination of the optimal sequence of additively distinct numbers., EUROPEAN SCIENTIFIC DISCUSSIONS. Proceedings of X International Scientific and Practical Conference Rome, Italy, 2021, August 15–17, p. 75–79.

4. Pavlenko, O., Tymoshenko A., Melnyk I., Luntovskyy A. et al. Searching Extreme Paths Based on Travelling Salesman's Problem for Wireless Emerging Networking, in Springer LNEE 965. Emerging Networking in the Digital Transformation Age: Approaches, Protocols, Platforms, Best Practices, and Energy Efficiency, by Springer LNEE 2023, issue 965, Springer Nature Cham, 24 p. (chapter #16).

5. Pavlenko, O.Yu., Timoshenko, A.H., Bondarenko, V.M. (2017). Algorithm for searching a series of sequentially additively distinct numbers. Bulletin of the University "Ukraine" Computer Science and Cybernetics., № 1 (18). P. 118–121.

**УДК 004.056 (083.94)**
**DOI https://doi.org/10.36994/2788-5518-2023-02-06-09**

# ПИТАННЯ ВИКОРИСТАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У СКЛАДІ БРОКЕРА ОБ'ЄКТНИХ ЗАПИТАНЬ В АРХІТЕКТУРІ СЕРВІС-ОРІЄНТОВАНОЇ СИСТЕМИ

***Рогоза В. С.,*** *д-р техн. наук, професор, Західнопоморський технологічний університет, м. Щецин, Польща; Навчально-науковий комплекс «Інститут прикладного системного аналізу – ННК ІПСА», Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського, м. Київ, Україна*
*https://orcid.org/0000-0003-2327-156X, wrogoza@wi.zut.edu.pl*

***Зарічний Я. С.,*** *Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», м. Київ, Україна*
*https://orcid.org/0000-0002-7596-5857, yarik120700@gmail.com*

***Анотація.*** *У статті розглядається питання підвищення ефективності реалізації послуг у сервісно-орієнтованій архітектурі (SOA) шляхом інтеграції рекомендаційних систем до складу брокера об'єктних запитань. Автори аналізують ключові проблеми, що виникають під час функціонування двох компонентів SOA: шаблону виявлення послуг на боці сервера та шаблону використання реєстру послуг.*

*Для шаблону виявлення послуг визначено проблеми складності налаштування балансувальника, необхідності індивідуального балансування трафіку через гетерогенність сервісів, труднощі балансування навантаження на брокер, динамічності процесів обробки запитів, а також складності виконання моніторингу й аналізу запитів для виявлення аномалій і прогнозування навантаження.*

*Для шаблону використання реєстру послуг виявлено проблеми неструктурованості даних про сервіси, відсутності метаданих про їхні функціональні можливості, інтерфейси, доступність та якість, а також обмеження механізмів фільтрації сервісів, що ускладнює їх пошук і вибір.*

*Як рішення пропонується включити рекомендаційну систему до балансувальника навантаження та сервісу реєстрацій. Для балансувальника вона забезпечить оптимізацію розподілу трафіку, автоматичне виявлення аномалій, прогнозування навантаження, оптимізацію використання ресурсів та адаптивне управління процесами обробки запитів.*

*Для сервісу реєстрацій рекомендаційна підсистема дасть змогу динамічно оновлювати рекомендації відповідно до попиту, групувати користувачів за інтересами, надавати контекстно-чутливі рекомендації, збагачувати профілі користувачів, використовувати колаборативну фільтрацію та оцінювати якість рекомендацій на основі зворотного зв'язку.*

*Інтеграція рекомендаційних систем може значно підвищити ефективність, продуктивність і досвід користувачів SOA завдяки персоналізованим рекомендаціям релевантних сервісів, що відповідають їхнім конкретним потребам. Автори вважають це важливим кроком у розвитку сервісно-орієнтованої архітектури.*

***Ключові слова:*** *сервіс-орієнтована архітектура, рекомендаційна система, балансувальник навантаження, реєстр сервісів, персоналізація, адаптивне управління.*

# USING RECOMMENDER SYSTEMS AS PART OF A BROKER OF OBJECT QUESTIONS IN THE ARCHITECTURE OF A SERVICE-ORIENTED SYSTEM

***Valeriy Rogoza,*** *dr hab., prof., West Pomeranian University of Technology, Szczecin, Poland; Educational and Scientific Complex "Institute of Applied Systems Analysis" – ESC "IASA", The National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*
*https://orcid.org/0000-0003-2327-156X, wrogoza@wi.zut.edu.pl*

***Yaroslav Zarichnyi,*** *National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine*
*https://orcid.org/0000-0002-7596-5857, yarik120700@gmail.com*

***Abstract.*** *The paper examines the issue of improving the efficiency of service implementation in a service-oriented architecture (SOA) by integrating recommendation systems into the object request broker. The authors analyze the key problems that arise during the operation of two components of SOA: the service discovery pattern on the server side and the service registry usage pattern.*

*For the service discovery pattern, the identified problems include the complexity of configuring the load balancer, the need for individual traffic balancing due to service heterogeneity, difficulties in balancing the*

*load on the broker, the dynamic nature of request processing, as well as the complexity of monitoring and analyzing requests for anomaly detection and load forecasting.*

*For the service registry usage pattern, the authors identified problems with unstructured data about services, lack of metadata about their functional capabilities, interfaces, availability, and quality, as well as limitations in service filtering mechanisms, which complicates the search and selection of services.*

*As a solution, it is proposed to include a recommendation system in the load balancer and service registry. For the load balancer, it will provide traffic distribution optimization, automatic anomaly detection, load forecasting, resource utilization optimization, and adaptive management of request processing processes.*

*For the service registry, the recommendation subsystem will allow dynamic updating of recommendations based on demand, grouping users by interests, providing context-sensitive recommendations, enriching user profiles, using collaborative filtering, and evaluating recommendation quality based on user feedback.*

*The integration of recommendation systems can significantly improve the efficiency, performance, and user experience of SOA through personalized recommendations of relevant services that meet their specific needs. The authors consider this an important step in the development of service-oriented architecture.*

***Key words:*** *service-oriented architecture; recommendation system; load balancing; service registry; personalization; adaptive management.*

**Вступ**

У світі, насиченому інформацією та вибором, рекомендації, що базуються на контенті, стають важливими для індивідуалізованого досвіду. Рекомендаційні системи виникають як ключове рішення в цьому контексті. Спочатку сприйняті як інструменти, що збирають і подають пропозиції, вони поступово перетворилися на комплексні програмні продукти, які використовують різні методології для надання рекомендацій щодо використання програмних застосунків, які найбільшою мірою відповідають вимогам користувачів і змісту вирішуваних задач. У сервісно-орієнтованій архітектурі (англ. service-oriented architecture, SOA) такі застосунки класифікуються як вебсервіси, призначені для сприяння створенню розподілених застосунків із швидкою, економічно ефективною та простою композицією у гетерогенних середовищах [1]. Ці програмні системи розроблені для надання функцій, доступних через вебтехнології з дотриманням встановлених для них стандартів і врахуванням архітектури систем сервісно-орієнтованих обчислень (англ. service-oriented computing, SOC).

Пошук і відбір документів та методів серед їх великої і постійно зростаючої кількості, які найбільш повно відповідають запитам користувачів, нерідко виявляється складним завданням. У сервісно-орієнтованих обчисленнях використовуються вебсервіси як основа для розробки розподілених застосунків з можливою їх композицією. Вебсервіси стають стандартизованими засобами для взаємодії між програмними застосунками різних платформ і структур. Незважаючи на критично важливу роль вебсервісів у SOA, існують проблеми з їх вибором і виявленням через обмеження в стандартах. До того ж процеси вибору й використання веб-служб ускладняються внаслідок певних недоліків у структурі реєстру UDDI та в мовах опису вебсервісів, таких як описи WSDL [1].

У цій статті виконано аналіз практичних викликів (названих у статті «проблемами»), що виникають під час експлуатації систем сервісно-орієнтованих обчислень, і переваг, які забезпечуються включенням спеціалізованої рекомендаційної системи у складі брокера об'єктних запитань в моделі SOA. Аналіз виконано з посиланнями на відомі ключові властивості рекомендаційних систем, які доводять доцільність їх використання в контексті брокерів об'єктних запитань для покращення функціональних можливостей сервіс-орієнтованих обчислень.

**Можливі варіанти включення рекомендаційних систем до складу брокера об'єктних запитань у SOA**

На нашу думку, ефективність реалізації послуг у SOA можна підвищити через накладання на компоненти брокера об'єктних запитань SOA додаткових функцій управління, моніторингу та прогнозування навантажень, які можна розглядати як функції спеціалізованих рекомендаційних систем. До такого висновку можна дійти, аналізуючи певні виклики, які виникають під час функціонування двох компонентів SOA: шаблону виявлення послуг на боці сервера та шаблону використання реєстру служб SOA

**Шаблон виявлення послуг на боці сервера**

Сервісні хореографії – це універсальний підхід до побудови розподілених систем на основі сервісів. У літературі можна знайти багато підходів, що стосуються різних аспектів хореографії обслуговування, таких як реалізованість хореографії та перевірка відповідності, розподілена координація, формальні хореографічні мови та масштабованість [2]. Як відомо, шаблон виявлення послуг на боці сервера, який є складником стандартної SOA, працює таким чином: клієнт ініціює запит до служби через балансувальника навантаження. Балансувальник навантаження, зі свого боку, запитує реєстр служби та спрямовує кожен запит до доступного екземпляра служби [3]. Так само, як виявляються замовлення на послуги на боці клієнта, екземпляри служби проходять процеси реєстрації та скасування реєстрації в реєстрі послуг.

Використання шаблону пошуку послуг у традиційній SOA викликає певні труднощі та проблеми в реалізації функцій, покладених на системи сервісно-орієнтованих обчислень. Визначимо деякі з них.

Складність налаштування. У середовищі сервісно-орієнтованих обчислень використовуються різні типи сервісів із різними характеристиками та вимогами до навантаження. Налаштування балансувальника навантаження вимагає уважного аналізу цих характеристик і встановлення правильних параметрів для оптимального розподілу трафіку між сервісами. Однак з урахуванням динамічної природи середовища SOA налаштування може стати складним завданням, оскільки потребує постійного моніторингу й адаптації до змін [4].

Необхідність застосування індивідуального підходу до балансування трафіку обробки об'єктних запитань. Ця необхідність обумовлена гетерогенністю сервісів. У SOA можуть працювати різні типи сервісів, такі як мікрослужби, монолітичні застосунки, сервери баз даних тощо. Кожен із цих сервісів може мати власні характеристики, такі як пропускна здатність, час відповіді, завантаженість тощо, які потребують індивідуального підходу до балансування трафіку [5].

Необхідність подолання проблем балансування навантаженням брокера. Підтримка масштабування є важливою функцією балансувальника навантаження в SOA. Зі збільшенням обсягу трафіку або кількості сервісів балансувальник повинен забезпечувати рівномірне розподілення навантаження й уникати перевантаження окремих елементів інфраструктури. Зазначені проблеми є особливо нагальними під час обробки дуже великих обсягів запитів і сервісів, які потенційно можуть відповідати вимогам цих запитів. Вирішення цієї проблеми напряму пов'язане з можливостями брокера передбачати надходження до системи нових запитів. На жаль, в існуючих SOA відсутні програмні засоби автоматичного балансування навантаження.

Труднощі, обумовлені динамічністю процесів обробки запитань. У середовищі SOA сервіси можуть бути додані, змінені або видалені в будь-який момент. Балансувальник навантаження повинен швидко реагувати на запитання, які надходять до системи від користувачів, і адаптуватися до нової конфігурації сервісів, тому автоматизація процесів розгортання та масштабування сервісів вимагає впровадження ефективних програмних засобів планування виконання запитів.

Складність виконання моніторингу й аналізу запитань. Балансувальник навантаження повинен постійно здійснювати моніторинг навантаження на всіх сервісах у SOA. Моніторинг передбачає оцінку швидкості відповіді, обсягу запитів та стану ресурсів і дає змогу вчасно виявляти перевантаження та реагувати на них. Поряд із моніторингом балансувальник навантаження повинен аналізувати трафік, який проходить через нього. Реалізація згаданих функцій має бути спрямована на виконання аналізу типів запитів і динаміки зміни обсягу трафіку, а також виявлення несправностей або аномальної активності окремих сервісів. На основі зібраної статистики балансувальник навантаження може бути в стані робити прогнози щодо майбутнього навантаження. Це дає змогу підготуватися до очікуваного зростання трафіку та вчасно приймати заходи для масштабування інфраструктури. Отже, для ефективного моніторингу й аналізу бажано використовувати автоматизовані засоби. Це можуть бути системи моніторингу й управління, які автоматично збирають дані про стан системи, виявляють аномалії та надсилають сповіщення в разі виникнення проблем. Для більш глибокого аналізу трафіку та виявлення тенденцій доцільно інтегрувати балансувальник навантаження із засобами аналізу даних, такими як, скажімо, системи BI (англ. Business Intelligence) або аналітичні інструменти, спрямовані на подолання зазначених проблем. Це дає змогу отримувати додаткові інсайти та робити точніші прогнози. На основі зібраних даних балансувальник навантаження може оптимізувати використання ресурсів, наприклад розподіляти трафік таким чином, щоб мінімізувати використання обчислювальних ресурсів або максимізувати швидкість відповіді для користувачів [3, 6].

Таким чином, можна стверджувати, що існує необхідність включення рекомендаційної системи до складу балансувальника, до функцій якої входитиме реалізація обчислювальних процесів, скерованих на подолання наслідків прояви згаданих вище проблем.

З огляду на можливості рекомендаційних систем [5] додавання рекомендаційної системи до складу балансувальника навантаження дасть змогу покласти на неї зазначені нижче функції.

Необхідність оптимізації розподілу трафіку. Рекомендаційна система може аналізувати різноманітні параметри, такі як пропускна здатність, завантаженість, час відповіді та стан ресурсів кожного сервісу. На основі цього аналізу вона може виробляти індивідуальні рекомендації для кожного запиту, враховуючи його характеристики та вимоги. Рекомендаційна система може використовувати дані про структуру SOA та взаємозв'язків між різними сервісами для оптимізації розподілу трафіку.

Доцільність автоматичного виявлення аномалій у викликах сервісів. Рекомендаційна система може використовувати різні методи аналізу, такі як статистичні моделі, машинне навчання й аналіз великих даних, для виявлення аномальної активності брокера запитань. Це дає змогу виявляти несподівані зміни в патернах трафіку, несправності в роботі сервісів або спроби злому системи. Крім того, на рекомендаційну систему доцільно покласти функції відслідковування подій і сповіщення про аномалії поведінки брокера, що дає можливість операторам швидко реагувати на них.

Необхідність прогнозування навантаження. Рекомендаційна система цілком здатна використовувати дані з історії трафіку для розробки моделей прогнозування майбутнього навантаження. Вона може враховувати фактори, які впливають на зміну трафіку, такі як день тижня, час доби, сезонність або маркетингові акції, унаслідок виконання цих дій рекомендаційна система набуває здатності регулярно оновлювати свої прогнози на основі нових даних і змін у середовищі.

Необхідність оптимізації використання ресурсів. Рекомендаційна система може аналізувати доступні ресурси та їхні обмеження, такі як обсяг пам'яті, кількість ядер процесора або пропускна здатність мережі, і рекомендувати оптимальні параметри конфігурації балансувальника навантаження, щоб максимально використовувати доступні ресурси й уникнути перевантаження.

Необхідність адаптивного управління процесами обробки запитань. Рекомендаційна система може використовувати зворотний зв'язок для оцінки ефективності своїх рекомендацій та адаптації до змін у середовищі. Це дає можливість автоматично коригувати стратегії розподілу трафіку та параметри конфігурації на основі отриманих результатів. Функції адаптації управління передбачають застосування процедур навчання, яке відбувається на основі досвіду, і таким чином можна поступово покращувати продуктивність та ефективність управління балансувальником навантаження.

**Шаблон використання реєстру служб**

Ключову роль у виявленні служб SOA відіграє реєстр служб, функціонуючи як база даних, що містить мережеві розташування примірників служб. Забезпечення високої доступності й ефективності пошуку служб, які відповідають запитам клієнтів, у режимі реального часу має вирішальне значення для ефективного функціонування реєстру послуг. Хоча клієнти можуть кешувати мережеві розташування, отримані з реєстру, ця інформація неминуче застаріває, що призводить до того, що клієнти не завжди можуть виявити потрібні їм екземпляри служб. Отже, реєстр послуг містить кластер серверів, які використовують протокол реплікації для підтримки узгодженості [4]. Використання існуючого шаблону часто викликає зазначені нижче проблеми.

*Неструктурованість даних.* Ця проблема виникає, коли інформація про послуги в сервісі реєстрацій не є структурованою або не відповідає певним стандартам. Різні сервіси можуть бути зареєстровані в реєстрі з використанням різних форматів даних. Наприклад, один сервіс може мати одну структуру метаданих або формат опису, а другий сервіс – іншу. Унаслідок такої невідповідності ускладнюється процес аналізу й обробки цих даних. Крім того, неструктурованість даних ускладнює їх пошук, аналіз та інтерпретацію. Це може призводити до втрати часу й ресурсів на пошук потрібної інформації та її подальший аналіз, а неструктурованість даних може призводити до помилок і неточностей у їхньому аналізі та використанні. Наприклад, відсутність стандартизованого формату може призвести до неправильного розуміння метаданих та їхнього невірного використання.

*Відсутність метаданих.* Однією з ключових проблем є відсутність або недостатня повнота метаданих про сервіси. Метадані можуть містити інформацію про функціональні можливості, інтерфейси, доступність, якість та інші характеристики сервісів. Брак цих даних викликає ускладнення у пошуку, виборі та використання сервісів. Метадані про функціональні можливості сервісів допомагають користувачам розуміти, які конкретні функції або послуги надає кожен сервіс. Метадані містять опис конкретних операцій, які можуть бути виконані, а також опис параметрів, потрібних для виконання цих операцій та значень, повернених після обробки запитів.

З описом метаданих пов'язана проблема побудови інтерфейсів. Використання відповідним чином побудованих інтерфейсів сприяє більш точній інтерпретації вимог користувачів до сервісів і сервісів між собою. Така інформація може містити формати даних, протоколи комунікації та доступні способи виклику сервісу. Метадані про доступність сервісів вказують на те, коли і за яких умов сервіс може бути доступний для використання. Зокрема, опис метаданих доцільно супроводжувати інформацію про час доступності, обмеження на кількість одночасних викликів або потребу автентифікації, а інформація про характеристики сервісів дає змогу оцінити їхню ефективність, надійність та швидкодію. Такі відомості можуть містити метрики продуктивності, рівень сервісу, обмеження на обробку даних та інші характеристики [6].

*Проблема пошуку та вибору сервісів.* Ця проблема виникає в сервісно-орієнтованих обчисленнях унаслідок у разі відсутності ефективних механізмів фільтрування сервісів. Наприклад, відсутність можливості фільтрувати сервіси за типом, функціональністю або рівнем доступності може призвести до труднощів у пошуку сервісів, що задовольняють конкретним вимогам користувача. Без можливості сортувати й ранжувати сервіси за певними критеріями користувачам нерідко важко зорієнтуватися серед великої кількості результатів пошуку. Скажімо, користувачам може бути потрібно сортувати сервіси за рівнем якості, швидкодією або надійністю, щоб знайти найкращий сервіс для своїх потреб. Якщо інформація про сервіси є неповною або взагалі відсутня, то пошук відповідних сервісів стає значною мірою непередбачуваним навіть у разі, коли користувач точно визначає вимоги до послуг, які йому потрібні. Наприклад, якщо в метаданих відсутні повні відомості про функціональність або параметри сервісів, користувачам буде важко оцінити, чи вони відповідають їхнім потребам [7, 8].

Крім того, якщо дані про сервіси в реєстрі не представлені у формі однорідної структури або формату, це ускладнює їх оброблення та порівняння користувачами навіть у випадках, коли опис вимог до сервісів чітко визначений користувачем і дані про параметри сервісів чітко визначені у відповідних описах сервісів.

Подолання окреслених вище труднощів може бути покладено на спеціалізовану певним чином рекомендаційну систему з додатковими функціями і не вимагатиме реконструкції вже існуючих SOA. На підставі наведених міркувань можна запропонувати модель SOA, у якій рекомендаційна система входить як підсистема сервісу реєстрацій послуг (рис. 1).

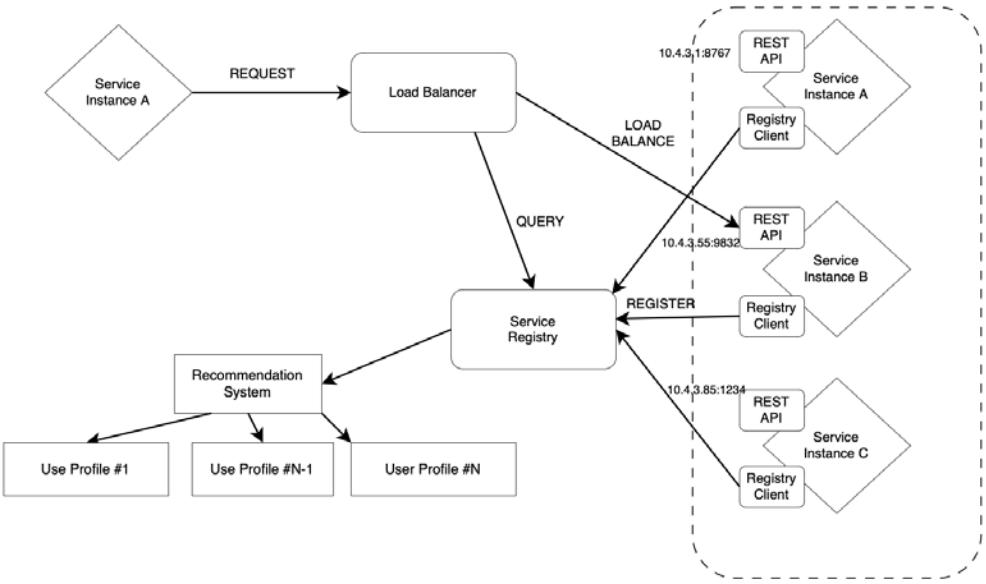


Рис. 1. Рекомендаційна система в складі сервісу реєстрацій послуг

Таким чином, поліпшення роботи сервісу реєстрацій завдяки застосуванню рекомендаційної підсистеми та створенню профілів користувачів може стати персоналізованим і сприяти підвищенню ефективності процесу пошуку та вибору сервісів з більш точним врахуванням потреб користувачів.

Можна виділити такі переваги запропонованої моделі, які випливають завдяки використанню рекомендаційної підсистеми.

*Динамічне оновлення рекомендацій.* Рекомендаційна підсистема може постійно аналізувати зміни в попиті на сервіси та динамічно оновлювати рекомендації відповідно до цих змін. Наприклад, якщо певний сервіс стає популярним серед користувачів, він може бути рекомендований іншим користувачам, які мають подібні вимоги.

*Групування користувачів за інтересами.* Використання рекомендаційної підсистеми може дати змогу створювати групи користувачів за їхніми інтересами та потребами. Наприклад, користувачі, які виявляють інтерес до певних типів сервісів, можуть бути об'єднані в спільні групи для зручності надання персоналізованих рекомендацій.

*Контекстно-чутливі рекомендації.* Рекомендаційна підсистема може враховувати контекстні фактори, такі як геолокація, час доби або пристрій, на якому виконується запит, для надання більш точних і релевантних рекомендацій. Наприклад, користувачам, які використовують мобільні пристрої, можуть бути рекомендовані послуги, оптимізовані саме з врахуванням згаданої мобільності пристроїв [9].

*Збагачення профілів користувачів.* Сервіс реєстрацій може збагачувати профілі користувачів за допомогою додаткових даних, таких як соціальні мережі, історія переглядів або попередні запити. Це дасть змогу створювати більш деталізовані та повні профілі, що поліпшить якість рекомендацій.

*Колаборативна фільтрація.* Використання колаборативної фільтрації дає змогу враховувати попередні вибори та поведінку користувачів, а також подібність їхніх виборів до інших користувачів. Цей підхід може сприяти формулюванню більш точних персоналізованих рекомендації [10].

*Оцінка якості рекомендацій.* Сервіс реєстрацій може мати механізми для оцінки ефективності рекомендацій з огляду на реакції користувачів на них. Наприклад, користувачі можуть залишати відгуки або оцінки після використання рекомендованих сервісів, що допомагає вдосконалювати якість рекомендацій в майбутньому.

**Висновки**

Використання рекомендаційної підсистеми в архітектурі SOC може значно поліпшити ефективність, продуктивність і користувацький досвід системи. Рекомендаційна підсистема допомагає користувачам швидко знаходити необхідні сервіси за допомогою персоналізованих рекомендацій, що відповідають їхнім уподобанням і потребам. Користувачі можуть отримувати рекомендації щодо найбільш підходящих сервісів для їхніх конкретних завдань або сценаріїв використання. За допомогою рекомендаційної підсистеми можуть створюватися персоналізовані профілі користувачів і надаватися рекомендації, які враховують їхні індивідуальні вподобання, історію взаємодій та контекст.

Шляхом рекомендацій користувачі можуть знаходити нові для себе сервіси або використовувати додаткові функціональності, які можуть бути корисними для їхніх поточних потреб. Рекомендаційні

системи можуть також допомогти користувачам вибирати найновіші й найактуальніші версії сервісів, що дає змогу підтримувати систему в сучасному стані. Таким чином, завдяки використанню рекомендаційної підсистеми у SOA можна підвищити зручність, ефективність і користувацькі вимоги від використання сервісів, що є певним кроком у розвитку цієї архітектури.

### *Література*

1. Fabrizio Ciancetta, Giovanni Bucci, Carmine Landi. Taxonomies of distributed measurement systems via UDDI registry. IEEE Instrumentation and Measurement Technology Conference, 2008, pp. 1316–1321.

2. Filippone G., Pompilio C., Autili M. & Tivoli M. An architectural style for scalable choreography-based microservice-oriented distributed systems. Computing, 105, 2023, pp. 1933–1956.

3. Chris Richardson, Service Discovery in a Microservices Architecture, 2015. URL: https://www.nginx.com/blog/service-discovery-in-a-microservices-architecture/ (дата звернення 20.02.2024).

4. Simone Cusimano, Service Discovery in Microservices, 2022. URL: https://www.baeldung.com/cs/service-discovery-microservices (дата звернення 15.02.2024).

5. Karolina Holewa, Perks of recommendation systems in business, 2023. URL: https://www.miquido.com/blog/perks-of-recommendation-systems-in-business/ (дата звернення 20.02.2024).

6. An SOA Design Patterns Recommendation System Based on Ontology URL: https://link.springer.com/chapter/10.1007/978-3-030-16657-1_95 (дата звернення 01.03.2024).

7. Ezdehar Jawabreh, Adel Taweel. Time-Aware QoS Web Service Selection Using Collaborative Filtering. A Literature Review, pp. 55–69.

8. Dang Y. Personalized Recommendation of Literature Resources in University Library Based on Abstract Content Filtering Algorithm, 2023.

9. Felfernig A., Atas M., Helic D., Tran T. N. T., Stettinger M., & Samer R. Algorithms for Group Recommendation. In Group Recommender Systems, 2024, pp. 29-61.

10. Trattner C., Said A., Boratto L., Felfernig A. Evaluating Group Recommender Systems. In Group Recommender Systems, 2024, pp. 63–75.

### *Reference*

1. Fabrizio Ciancetta, Giovanni Bucci, Carmine Landi (2008). Taxonomies of distributed measurement systems via UDDI registry, IEEE Instrumentation and Measurement Technology Conference, pp. 1316–1321.

2. Filippone, G., Pompilio, C., Autili, M., & Tivoli, M. (2023). An architectural style for scalable choreography-based microservice-oriented distributed systems. Computing, 105, pp. 1933–1956.

3. Chris Richardson (2015). Service Discovery in a Microservices Architecture [Online]. Available: https://www.nginx.com/blog/service-discovery-in-a-microservices-architecture/.

4. Simone Cusimano (2022). Service Discovery in Microservices [Online]. Available: https://www.baeldung.com/cs/service-discovery-microservices.

5. Karolina Holewa (2023). Perks of recommendation systems in business [Online]. Available: https://www.miquido.com/blog/perks-of-recommendation-systems-in-business/.

6. An SOA Design Patterns Recommendation System Based on Ontology [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-16657-1_95.

7. Ezdehar Jawabreh, Adel Taweel. Time-Aware QoS Web Service Selection Using Collaborative Filtering. A Literature Review, pp. 55–69.

8. Dang, Y. (2023). Personalized Recommendation of Literature Resources in University Library Based on Abstract Content Filtering Algorithm.

9. Felfernig, A., Atas, M., Helic, D., Tran, T. N. T., Stettinger, M., & Samer, R. (2024). Algorithms for Group Recommendation. In Group Recommender Systems, pp. 29–61.

10. Trattner, C., Said, A., Boratto, L., Felfernig, A. (2024). Evaluating Group Recommender Systems. In Group Recommender Systems, pp. 63–75.

**НАУКОВЕ ВИДАННЯ**

# ІНФОКОМУНІКАЦІЙНІ ТА КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ

# INFOCOMMUNICATION AND COMPUTER TECHNOLOGIES

## № 2 (06) 2023

Підписано до друку: 29.12.2023.
Формат 60x84/8. Arial.
Папір офсет. Цифровий друк. Ум. друк. арк. 9,77. Замов. № 0524/342. Наклад 300 прим.

Видавництво і друкарня – Видавничий дім «Гельветика»
65101, Україна, м. Одеса, вул. Інглезі, 6/1
Телефон +38 (095) 934 48 28, +38 (097) 723 06 08
E-mail: mailbox@helvetica.ua
Свідоцтво суб'єкта видавничої справи
ДК № 7623 від 22.06.2022 р.